

# Projet Bee-Honey't

1.1

MELLAH Florentin  
ROSSI Enzo  
LAURAIN Clément

## Table des matières

<b>1</b>	<b>Bee-Honey't</b>	<b>1</b>
1.1	Table des matières . . . . .	1
<b>2</b>	<b>Changelog</b>	<b>2</b>
<b>3</b>	<b>Installation</b>	<b>10</b>
<b>4</b>	<b>README</b>	<b>10</b>
<b>5</b>	<b>A propos</b>	<b>15</b>
<b>6</b>	<b>Licence GPL</b>	<b>15</b>
<b>7</b>	<b>Liste des choses à faire</b>	<b>15</b>
<b>8</b>	<b>Documentation des espaces de nommage</b>	<b>16</b>
8.1	Paquetage fr . . . . .	16
8.2	Paquetage fr.campus . . . . .	16
8.3	Paquetage fr.campus.laurainc . . . . .	16
8.4	Paquetage fr.campus.laurainc.honeybee . . . . .	16
8.5	Référence de l'espace de nommage Ui . . . . .	17
<b>9</b>	<b>Documentation des classes</b>	<b>17</b>
9.1	Référence de la classe Alertes . . . . .	17
9.1.1	Description détaillée . . . . .	19
9.1.2	Documentation des constructeurs et destructeur . . . . .	19
9.1.3	Documentation des fonctions membres . . . . .	20
9.1.4	Documentation des données membres . . . . .	37
9.2	Référence de la classe fr.campus.laurainc.honeybee.alertesActivity . . . . .	41
9.2.1	Documentation des fonctions membres . . . . .	42
9.2.2	Documentation des données membres . . . . .	42
9.3	Référence de la classe BaseDeDonnees . . . . .	43
9.3.1	Documentation des constructeurs et destructeur . . . . .	44
9.3.2	Documentation des fonctions membres . . . . .	45

9.3.3	Documentation des données membres . . . . .	51
9.4	Référence de la classe <code>fr.campus.laurainc.honeybee.BaseDeDonnees</code> . . . . .	53
9.4.1	Description détaillée . . . . .	54
9.4.2	Documentation des constructeurs et destructeur . . . . .	54
9.4.3	Documentation des fonctions membres . . . . .	56
9.4.4	Documentation des données membres . . . . .	62
9.5	Référence de la classe <code>fr.campus.laurainc.honeybee.CarteActivity</code> . . . . .	64
9.5.1	Documentation des fonctions membres . . . . .	64
9.5.2	Documentation des données membres . . . . .	65
9.6	Référence de la classe <code>fr.campus.laurainc.honeybee.ClientMQTT</code> . . . . .	66
9.6.1	Documentation des constructeurs et destructeur . . . . .	67
9.6.2	Documentation des fonctions membres . . . . .	68
9.6.3	Documentation des données membres . . . . .	75
9.7	Référence de la classe <code>Communication</code> . . . . .	77
9.7.1	Description détaillée . . . . .	78
9.7.2	Documentation des constructeurs et destructeur . . . . .	78
9.7.3	Documentation des fonctions membres . . . . .	79
9.7.4	Documentation des données membres . . . . .	82
9.8	Référence de la classe <code>fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySql</code> . . . . .	83
9.8.1	Documentation des fonctions membres . . . . .	83
9.8.2	Documentation des données membres . . . . .	85
9.9	Référence de la classe <code>fr.campus.laurainc.honeybee.DashboardActivity</code> . . . . .	86
9.9.1	Documentation des fonctions membres . . . . .	87
9.9.2	Documentation des données membres . . . . .	94
9.10	Référence de la classe <code>fr.campus.laurainc.honeybee.DetailsRucheActivity</code> . . . . .	97
9.10.1	Description détaillée . . . . .	97
9.10.2	Documentation des fonctions membres . . . . .	97
9.11	Référence de la structure <code>DonneesBatterie</code> . . . . .	98
9.11.1	Description détaillée . . . . .	99
9.11.2	Documentation des données membres . . . . .	99
9.12	Référence de la structure <code>DonneesRuche</code> . . . . .	100

9.12.1	Description détaillée	101
9.12.2	Documentation des données membres	101
9.13	Référence de la classe <code>fr.campus.laurainc.honeybee.ExampleInstrumentedTest</code>	103
9.13.1	Description détaillée	104
9.13.2	Documentation des fonctions membres	104
9.14	Référence de la classe <code>fr.campus.laurainc.honeybee.ExampleUnitTest</code>	104
9.14.1	Description détaillée	105
9.14.2	Documentation des fonctions membres	105
9.15	Référence de la classe <code>fr.campus.laurainc.honeybee.GraphActivity</code>	106
9.15.1	Documentation des fonctions membres	107
9.15.2	Documentation des données membres	113
9.16	Référence de la classe <code>fr.campus.laurainc.honeybee.homeActivity</code>	117
9.16.1	Documentation des fonctions membres	118
9.16.2	Documentation des données membres	119
9.17	Référence de la classe <code>fr.campus.laurainc.honeybee.HoneyBee</code>	120
9.17.1	Description détaillée	120
9.17.2	Documentation des données membres	120
9.18	Référence de la classe <code>InfosBatterie</code>	122
9.18.1	Description détaillée	123
9.18.2	Documentation des constructeurs et destructeur	123
9.18.3	Documentation des fonctions membres	124
9.18.4	Documentation des données membres	127
9.19	Référence de la classe <code>InfosEnsoleillement</code>	128
9.19.1	Description détaillée	129
9.19.2	Documentation des constructeurs et destructeur	129
9.19.3	Documentation des fonctions membres	129
9.19.4	Documentation des données membres	132
9.20	Référence de la classe <code>InfosHumidite</code>	133
9.20.1	Description détaillée	135
9.20.2	Documentation des constructeurs et destructeur	135
9.20.3	Documentation des fonctions membres	136

9.20.4	Documentation des données membres	141
9.21	Référence de la classe InfosPoids	143
9.21.1	Description détaillée	144
9.21.2	Documentation des constructeurs et destructeur	144
9.21.3	Documentation des fonctions membres	144
9.21.4	Documentation des données membres	146
9.22	Référence de la classe InfosPressionAtmospherique	147
9.22.1	Description détaillée	148
9.22.2	Documentation des constructeurs et destructeur	148
9.22.3	Documentation des fonctions membres	148
9.22.4	Documentation des données membres	151
9.23	Référence de la classe InfosTemperature	152
9.23.1	Description détaillée	154
9.23.2	Documentation des constructeurs et destructeur	155
9.23.3	Documentation des fonctions membres	155
9.23.4	Documentation des données membres	161
9.24	Référence de la classe fr.campus.laurainc.honeybee.MainActivity	162
9.24.1	Description détaillée	164
9.24.2	Documentation des fonctions membres	164
9.24.3	Documentation des données membres	165
9.25	Référence de la structure MesureHoraireEnvironnement	167
9.25.1	Description détaillée	168
9.25.2	Documentation des données membres	168
9.26	Référence de la structure MesureHoraireRuche	170
9.26.1	Description détaillée	171
9.26.2	Documentation des données membres	171
9.27	Référence de la structure MesuresHoraireEnsoleillement	172
9.27.1	Description détaillée	173
9.27.2	Documentation des données membres	173
9.28	Référence de la classe fr.campus.laurainc.honeybee.NouvelleRucheActivity	174
9.28.1	Documentation des fonctions membres	175

9.28.2	Documentation des données membres	177
9.29	Référence de la classe NouvelleRuchelhm	180
9.29.1	Description détaillée	182
9.29.2	Documentation des constructeurs et destructeur	182
9.29.3	Documentation des fonctions membres	183
9.29.4	Documentation des données membres	184
9.30	Référence de la classe fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity	185
9.30.1	Description détaillée	186
9.30.2	Documentation des fonctions membres	186
9.30.3	Documentation des données membres	187
9.31	Référence de la classe QDialog	188
9.32	Référence de la classe QObject	188
9.33	Référence de la classe QWidget	188
9.34	Référence de la classe ReglagesAlertesIhm	189
9.34.1	Description détaillée	190
9.34.2	Documentation des constructeurs et destructeur	190
9.34.3	Documentation des fonctions membres	191
9.34.4	Documentation des données membres	195
9.35	Référence de la classe fr.campus.laurainc.honeybee.Ruche	196
9.35.1	Documentation des constructeurs et destructeur	198
9.35.2	Documentation des fonctions membres	200
9.35.3	Documentation des données membres	223
9.36	Référence de la classe Ruche	231
9.36.1	Description détaillée	233
9.36.2	Documentation des constructeurs et destructeur	233
9.36.3	Documentation des fonctions membres	235
9.36.4	Documentation des données membres	260
9.37	Référence de la classe Ruchelhm	263
9.37.1	Description détaillée	265
9.37.2	Documentation des constructeurs et destructeur	265
9.37.3	Documentation des fonctions membres	266
9.37.4	Documentation des données membres	292
9.38	Référence de la classe fr.campus.laurainc.honeybee.TimeAsXAxisLabelFormatter	295
9.38.1	Documentation des constructeurs et destructeur	295
9.38.2	Documentation des fonctions membres	295
9.38.3	Documentation des données membres	296

<b>10 Documentation des fichiers</b>	<b>296</b>
10.1 Référence du fichier alertes.cpp	296
10.1.1 Description détaillée	296
10.2 Référence du fichier alertes.h	296
10.2.1 Description détaillée	297
10.3 Référence du fichier alertesActivity.java	297
10.4 Référence du fichier baseDeDonnees.cpp	297
10.4.1 Description détaillée	297
10.5 Référence du fichier baseDeDonnees.h	297
10.5.1 Description détaillée	298
10.6 Référence du fichier BaseDeDonnees.java	298
10.7 Référence du fichier CarteActivity.java	298
10.8 Référence du fichier Changelog.md	298
10.9 Référence du fichier ClientMQTT.java	298
10.10 Référence du fichier communication.cpp	299
10.10.1 Description détaillée	299
10.11 Référence du fichier communication.h	299
10.11.1 Description détaillée	299
10.12 Référence du fichier DashboardActivity.java	299
10.13 Référence du fichier DetailsRucheActivity.java	300
10.14 Référence du fichier ExampleInstrumentedTest.java	300
10.15 Référence du fichier ExampleUnitTest.java	300
10.16 Référence du fichier GraphActivity.java	300
10.17 Référence du fichier homeActivity.java	300
10.18 Référence du fichier HoneyBee.java	301
10.19 Référence du fichier infosBatterie.cpp	301
10.19.1 Description détaillée	301
10.20 Référence du fichier infosBatterie.h	301
10.20.1 Description détaillée	302
10.21 Référence du fichier infosEnsoleillement.cpp	302
10.21.1 Description détaillée	302

10.22	Référence du fichier infosEnsoleillement.h . . . . .	302
10.22.1	Description détaillée . . . . .	302
10.23	Référence du fichier infosHumidite.cpp . . . . .	303
10.23.1	Description détaillée . . . . .	303
10.24	Référence du fichier infosHumidite.h . . . . .	303
10.24.1	Description détaillée . . . . .	303
10.25	Référence du fichier infosPoids.cpp . . . . .	303
10.25.1	Description détaillée . . . . .	304
10.26	Référence du fichier infosPoids.h . . . . .	304
10.26.1	Description détaillée . . . . .	304
10.27	Référence du fichier infosPressionAtmospherique.cpp . . . . .	304
10.27.1	Description détaillée . . . . .	304
10.28	Référence du fichier infosPressionAtmospherique.h . . . . .	305
10.28.1	Description détaillée . . . . .	305
10.29	Référence du fichier infosTemperature.cpp . . . . .	305
10.29.1	Description détaillée . . . . .	305
10.30	Référence du fichier infosTemperature.h . . . . .	305
10.30.1	Description détaillée . . . . .	306
10.31	Référence du fichier INSTALL.md . . . . .	306
10.32	Référence du fichier main.cpp . . . . .	306
10.32.1	Description détaillée . . . . .	306
10.32.2	Documentation des fonctions . . . . .	306
10.33	Référence du fichier MainActivity.java . . . . .	307
10.34	Référence du fichier NouvelleRucheActivity.java . . . . .	307
10.35	Référence du fichier nouvelleRuchelhm.cpp . . . . .	307
10.35.1	Description détaillée . . . . .	308
10.36	Référence du fichier nouvelleRuchelhm.h . . . . .	308
10.37	Référence du fichier parametres.h . . . . .	308
10.37.1	Description détaillée . . . . .	309
10.37.2	Documentation des macros . . . . .	309
10.37.3	Documentation du type de l'énumération . . . . .	312



10.38	Référence du fichier ParametresHoneyBeeActivity.java	314
10.39	Référence du fichier README.md	314
10.40	Référence du fichier reglagesAlertesIhm.cpp	314
10.40.1	Description détaillée	314
10.41	Référence du fichier reglagesAlertesIhm.h	314
10.41.1	Description détaillée	315
10.42	Référence du fichier ruche.cpp	315
10.42.1	Description détaillée	315
10.43	Référence du fichier ruche.h	316
10.43.1	Description détaillée	316
10.44	Référence du fichier Ruche.java	316
10.45	Référence du fichier ruchelhm.cpp	317
10.45.1	Description détaillée	317
10.46	Référence du fichier ruchelhm.h	317
10.46.1	Description détaillée	318
10.47	Référence du fichier TimeAsXAxisLabelFormatter.java	318

<b>Index</b>	<b>319</b>
--------------	------------

## 1 Bee-Honey't

Système autonome permettant de connaître à distance certains paramètres d'une ruche afin d'assurer son suivi et d'évaluer la santé des abeilles.

*Clément LAURAIN* [laurain.clement.contact@gmail.com](mailto:laurain.clement.contact@gmail.com)

*Florentin MELLAH* [florentinmellah@gmail.com](mailto:florentinmellah@gmail.com)

*Enzo ROSSI* [enzo.rossi13160contact@gmail.com](mailto:enzo.rossi13160contact@gmail.com)

### 1.1 Table des matières

- [README](#)
- [Changelog](#)
- [Installation](#)
- [Liste des choses à faire](#)
- [A propos](#)
- [Licence GPL](#)

## 2 Changelog

r124 | synapse | 2019-06-05 16 :29 :08 +0200 (mer. 05 juin 2019) | 1 ligne

intégration du restant de l'interface

r123 | erossi1 | 2019-06-05 00 :26 :26 +0200 (mer. 05 juin 2019) | 1 ligne

ajout de l'enregistrement des données Environnement et ensoleillement + représentation des moyennes journalières des données d'environnement + ensoleillement

r122 | erossi1 | 2019-06-04 18 :44 :24 +0200 (mar. 04 juin 2019) | 1 ligne

reagencement des courbes d'un tracé sur une heure par un tracé par un tracé par heure + ajout bdd 0.4 + ajout de la mesure moyenne mini maxi

r121 | synapse | 2019-06-04 16 :45 :09 +0200 (mar. 04 juin 2019) | 1 ligne

Ajout et suppression de classe, désabonnement au TTN

r120 | tvaira | 2019-06-04 06 :46 :36 +0200 (mar. 04 juin 2019) | 1 ligne

Déconnexion Serveur TTN

r119 | erossi1 | 2019-06-02 01 :10 :29 +0200 (dim. 02 juin 2019) | 1 ligne

ajout de la fonctionnalité de graphique (courbes) retracant les données en fonction du temps toutes les heures

r118 | erossi1 | 2019-05-29 16 :50 :34 +0200 (mer. 29 mai 2019) | 1 ligne

ajout d'onglet mesure et courbe dans ruche ihm + dans l'onglet courbe de ruche ajout 6 courbes correspondantes au 6 mesure reçue

r117 | synapse | 2019-05-29 16 :08 :59 +0200 (mer. 29 mai 2019) | 1 ligne

Intégration du TTN selon la ruche et fonctionnalité ajouter ruche dans BDD

r116 | fmellah | 2019-05-29 12 :24 :24 +0200 (mer. 29 mai 2019) | 1 ligne

écriture des données dans la bdd

r115 | tvaira | 2019-05-29 05 :54 :38 +0200 (mer. 29 mai 2019) | 1 ligne

Schema base de données v0.3

r114 | erossi1 | 2019-05-28 17 :33 :25 +0200 (mar. 28 mai 2019) | 1 ligne

ajout de l'insertion de seuil d'alerte par défaut lors de la création d'une ruche

r113 | erossi1 | 2019-05-28 15 :52 :58 +0200 (mar. 28 mai 2019) | 1 ligne

insertion des données correspondantes au fpor 3,4,5 dans la base de données

r112 | fmellah | 2019-05-28 15 :50 :43 +0200 (mar. 28 mai 2019) | 1 ligne

ajout des alertes ensoleillement, batterie, poids

r111 | erossi1 | 2019-05-28 13 :44 :03 +0200 (mar. 28 mai 2019) | 1 ligne

modification de la base de données en version 0.3 + modification des méthodes pour l'insertion dans la base de données

r110 | erossi1 | 2019-05-27 23 :08 :22 +0200 (lun. 27 mai 2019) | 1 ligne

ajout de methode de la classe [Ruche](#) permetnt l'insertion de données dans la base de données

r109 | erossi1 | 2019-05-27 17 :39 :19 +0200 (lun. 27 mai 2019) | 1 ligne

ajout de methode de cacul de moyenne de maximu et de minimum aux classes InfosTempatures(temperaturesExrterieur) [Infos↔Ensoleillement](#) et infosHumidite

r108 | tvaira | 2019-05-27 16 :44 :58 +0200 (lun. 27 mai 2019) | 1 ligne

Ajout module SimpleMail

r107 | tvaira | 2019-05-27 16 :40 :45 +0200 (lun. 27 mai 2019) | 2 lignes

Ajout du module fourni pour envoyer des mails

r106 | erossi1 | 2019-05-27 11 :55 :15 +0200 (lun. 27 mai 2019) | 1 ligne

ajout la methode calculerMoyennnePressionAtmospherique

r105 | tvaira | 2019-05-27 11 :09 :52 +0200 (lun. 27 mai 2019) | 2 lignes

Ajout des TODOs pour la gestion des seuils dans la BDD

r104 | tvaira | 2019-05-27 08 :12 :21 +0200 (lun. 27 mai 2019) | 1 ligne

Schema BDD 0.2

r103 | tvaira | 2019-05-27 08 :07 :10 +0200 (lun. 27 mai 2019) | 2 lignes

Détection changement d'heure pour calcul moyenne

r102 | tvaira | 2019-05-26 19 :20 :09 +0200 (dim. 26 mai 2019) | 2 lignes

Correction calcul moyenne température interieur

r101 | erossi1 | 2019-05-26 13 :49 :59 +0200 (dim. 26 mai 2019) | 1 ligne

Mise en place du traitement dans la methode traitementDonneeTemperatureInterieur

r100 | erossi1 | 2019-05-24 12 :00 :52 +0200 (ven. 24 mai 2019) | 1 ligne

poursuite de la methode de traitement de la tempereture interieur et exterieur dans infosTemperature

r99 | erossi1 | 2019-05-23 16 :25 :16 +0200 (jeu. 23 mai 2019) | 1 ligne

ajout des methodestraiterDonneesTemperatureExterieur et interieur dans la classe infosTemperature

r98 | synapse | 2019-05-23 16 :11 :44 +0200 (jeu. 23 mai 2019) | 1 ligne

Liste déroulante et requete sql et ttn personnalisé

r97 | erossi1 | 2019-05-23 10 :17 :35 +0200 (jeu. 23 mai 2019) | 1 ligne

ajout de la fonctionalite supprimer une ruche dans ruche ihm

r96 | erossi1 | 2019-05-23 09 :28 :03 +0200 (jeu. 23 mai 2019) | 1 ligne

ajout icon ensoleillement et pression

r95 | erossi1 | 2019-05-23 09 :07 :27 +0200 (jeu. 23 mai 2019) | 1 ligne

ajout icon poids et batterie

r94 | erossi1 | 2019-05-23 09 :02 :50 +0200 (jeu. 23 mai 2019) | 1 ligne

ajout icon poids et batterie

r93 | erossi1 | 2019-05-23 08 :52 :22 +0200 (jeu. 23 mai 2019) | 1 ligne

ajout icon poids et batterie

r92 | erossi1 | 2019-05-23 08 :49 :27 +0200 (jeu. 23 mai 2019) | 1 ligne

ajout icon poids et batterie

r91 | erossi1 | 2019-05-22 23 :42 :31 +0200 (mer. 22 mai 2019) | 1 ligne

ajout d'un slot pour la suppression d'une ruche dans ruchelhm

r90 | erossi1 | 2019-05-22 22 :42 :30 +0200 (mer. 22 mai 2019) | 1 ligne

ajout d'un bouton supprimer ruche dans ruchelhm

r89 | fmellah | 2019-05-21 17 :57 :33 +0200 (mar. 21 mai 2019) | 1 ligne

modification ihm

r88 | erossi1 | 2019-05-21 17 :41 :39 +0200 (mar. 21 mai 2019) | 1 ligne

ajout des images batterie

r87 | synapse | 2019-05-21 17 :38 :58 +0200 (mar. 21 mai 2019) | 1 ligne

Recuperation données TTN et BDD pour une ruche réalisé

r86 | erossi1 | 2019-05-21 17 :27 :03 +0200 (mar. 21 mai 2019) | 1 ligne

reagencement visuelle de rucheihm

r85 | fmellah | 2019-05-21 16 :05 :24 +0200 (mar. 21 mai 2019) | 1 ligne

ajout des classes infosBatterie et infosPoids + affichage poids et charge de la batterie

r84 | erossi1 | 2019-05-21 15 :59 :01 +0200 (mar. 21 mai 2019) | 1 ligne

ajout de la selection de lapplicationId dans nouvelleRuchelhm + refonte structurelle des des structure pour l'organistion des données des ruches dans [ruche.h](#) et [ruche.cpp](#)

r83 | fmellah | 2019-05-21 09 :59 :35 +0200 (mar. 21 mai 2019) | 1 ligne

ajout fonctionnalité de choix des ruches

r82 | tvaira | 2019-05-20 19 :46 :16 +0200 (lun. 20 mai 2019) | 2 lignes

Vérifications tag 0.2

r81 | tvaira | 2019-05-20 10 :34 :18 +0200 (lun. 20 mai 2019) | 1 ligne

Ajout documentation Doxygen pour tag 0.2

r80 | tvaira | 2019-05-20 10 :19 :36 +0200 (lun. 20 mai 2019) | 1 ligne

Passage des fichiers Doxygen au format Markdown

r79 | erossi1 | 2019-05-17 19 :13 :47 +0200 (ven. 17 mai 2019) | 1 ligne

reagencement visuelle de l'ihm nouvelleRuchelhm

r78 | erossi1 | 2019-05-17 18 :48 :16 +0200 (ven. 17 mai 2019) | 1 ligne

tag de la version 0.2

r77 | fmellah | 2019-05-17 13 :16 :08 +0200 (ven. 17 mai 2019) | 1 ligne

ajout de toutes les alertes

r76 | erossi1 | 2019-05-11 19 :47 :02 +0200 (sam. 11 mai 2019) | 1 ligne

ajout de la methode initialiserWidgets

r75 | erossi1 | 2019-05-11 18 :08 :19 +0200 (sam. 11 mai 2019) | 1 ligne

remediation du code pour le tag de la version 0.2

r74 | tvaira | 2019-05-11 12 :28 :55 +0200 (sam. 11 mai 2019) | 1 ligne

Ajout afficherLesMesures

r73 | erossi1 | 2019-05-09 16 :32 :08 +0200 (jeu. 09 mai 2019) | 1 ligne

ajout des widget pour la pression et l'ensoleillement

r72 | erossi1 | 2019-05-08 23 :02 :09 +0200 (mer. 08 mai 2019) | 1 ligne

ajout sur le serveur TTN le decodage de chaque mesure ce qui a pour impact la reception de celles ci grace a la methode recuperer↔  
Donnnes dans [Ruche](#)

r71 | tvaira | 2019-05-07 17 :21 :48 +0200 (mar. 07 mai 2019) | 2 lignes

Exemple extraction du poids

r70 | tvaira | 2019-05-07 14 :39 :49 +0200 (mar. 07 mai 2019) | 1 ligne

Exemple de décodage des données JSON du serveur TTN

r69 | erossi1 | 2019-05-06 00 :35 :23 +0200 (lun. 06 mai 2019) | 1 ligne

affichage de la pression atmospherique + ensolleimment en QDebug dans [Ruchelhm](#) TODO effectuer a l'aide de widget graphique

r68 | erossi1 | 2019-05-03 15 :51 :26 +0200 (ven. 03 mai 2019) | 1 ligne

modication de nouvelle-ruchelHM avec l'ajout de zones texte permettant la saisi du nom prenom email nom de topic + signaux et slot dans cette classes + slot de ception des donnees de nouvelle ruchelhm dans ruche + redefinition du constructeur de la classe [Ruche](#) avec l'ajout des attributs nom prenom email nomp de topic

r67 | synapse | 2019-05-03 12 :57 :37 +0200 (ven. 03 mai 2019) | 1 ligne

essai decodage trame

r66 | fmellah | 2019-05-03 12 :15 :07 +0200 (ven. 03 mai 2019) | 1 ligne

modification de l'ihm + ajout des alertes sur celle ci

r65 | erossi1 | 2019-05-02 16 :25 :53 +0200 (jeu. 02 mai 2019) | 1 ligne

commencement de l'intergration de la creation d'une nouvelle ruche

r64 | synapse | 2019-05-02 16 :25 :19 +0200 (jeu. 02 mai 2019) | 1 ligne

Ajout du TTN dans le projet

r63 | erossi1 | 2019-05-02 13 :35 :20 +0200 (jeu. 02 mai 2019) | 1 ligne

modifiacation visuelle de maNouvelleRuchelhm

r62 | fmellah | 2019-04-25 16 :42 :55 +0200 (jeu. 25 avril 2019) | 1 ligne

ajout des signaux/slot et des connect de la classe alerte vers les classes rucheihm et ruche

r61 | synapse | 2019-04-25 16 :26 :55 +0200 (jeu. 25 avril 2019) | 1 ligne

Recuperation des données depuis la base de donnees

r60 | fmellah | 2019-04-25 15 :47 :33 +0200 (jeu. 25 avril 2019) | 1 ligne

creation de la classe alertes

r59 | fmellah | 2019-04-25 15 :46 :58 +0200 (jeu. 25 avril 2019) | 1 ligne

creation de la classe alertes

r58 | erossi1 | 2019-04-25 15 :42 :30 +0200 (jeu. 25 avril 2019) | 1 ligne

ajout de toute les connnection clase infos vers rucher et ihm vers ruches

r57 | erossi1 | 2019-04-23 21 :57 :31 +0200 (mar. 23 avril 2019) | 1 ligne

creation de tout les slots de traitement dans les classes Infos et ajout de l'association avec l'horodatage

r56 | tvaira | 2019-04-06 10 :01 :47 +0200 (sam. 06 avril 2019) | 1 ligne

Ajout de la documentation pour le tag 0.1

r55 | erossi1 | 2019-04-05 11 :57 :48 +0200 (ven. 05 avril 2019) | 1 ligne

création du tag 0.1

r54 | erossi1 | 2019-04-05 11 :45 :13 +0200 (ven. 05 avril 2019) | 1 ligne

ajout de la documentation complet dioxygene

r53 | tvaira | 2019-04-02 16 :33 :36 +0200 (mar. 02 avril 2019) | 1 ligne

Correction problème compilation

r52 | fmellah | 2019-04-02 12 :26 :59 +0200 (mar. 02 avril 2019) | 1 ligne

ajout de todo pour l'ihm et la connection à la base de données

r51 | tvaira | 2019-04-02 05 :00 :20 +0200 (mar. 02 avril 2019) | 2 lignes

Validation des commentaires

r50 | erossi1 | 2019-04-01 22 :50 :56 +0200 (lun. 01 avril 2019) | 1 ligne

Amélioration de la doc doxygene TODO à poursuivre

r49 | erossi1 | 2019-03-31 20 :17 :06 +0200 (dim. 31 mars 2019) | 1 ligne

ajout d'une methode dans [Ruchelhm](#) initialiserThermometre + ajout de deux methodes dans [Ruche](#) recupereHrordatage + recupererDonnes + ajout de l'affichage de l'humidite interieur

r48 | tvaira | 2019-03-30 08 :41 :10 +0100 (sam. 30 mars 2019) | 1 ligne

Ajout des TODOs

r47 | tvaira | 2019-03-30 07 :58 :47 +0100 (sam. 30 mars 2019) | 1 ligne

Controle Qualité Revue 2

r46 | erossi1 | 2019-03-28 22 :45 :50 +0100 (jeu. 28 mars 2019) | 1 ligne

modification de la fonction extraire donnees et extraire numero port

r45 | synapse | 2019-03-28 22 :17 :42 +0100 (jeu. 28 mars 2019) | 1 ligne

Debut etude bdd

r44 | erossi1 | 2019-03-28 21 :56 :52 +0100 (jeu. 28 mars 2019) | 1 ligne

ajout de signaux et slot pour le traitement des donnee des mesure a l'interieur de la ruche

r43 | synapse | 2019-03-28 16 :34 :04 +0100 (jeu. 28 mars 2019) | 1 ligne

Suppression fichiers inutiles

r42 | erossi1 | 2019-03-28 15 :52 :17 +0100 (jeu. 28 mars 2019) | 1 ligne

modif du switch case

r41 | fmellah | 2019-03-28 15 :50 :51 +0100 (jeu. 28 mars 2019) | 1 ligne

ajout d'une fenetre de dialogue pour créer une nouvelle ruche

r40 | fmellah | 2019-03-28 13 :40 :47 +0100 (jeu. 28 mars 2019) | 1 ligne

modif ihm en prenant compte des exigences du client : ajout d'un affichage numerique pour les temperatures

r39 | erossi1 | 2019-03-28 12 :32 :49 +0100 (jeu. 28 mars 2019) | 1 ligne

ajout des attribut humiditeInterieur humiditeExterieur temperatureInterieur temperatureExterieur avec leurs mutateurs

r38 | fmellah | 2019-03-28 11 :34 :30 +0100 (jeu. 28 mars 2019) | 1 ligne

modficiation du nom du device du TTN + ajout du boutons pour créer une nouvelle ruche

r37 | erossi1 | 2019-03-27 13 :40 :06 +0100 (mer. 27 mars 2019) | 1 ligne

ajout des signaux nouvelle donnees dans la methode extraire donnees de la classe ruche

r36 | fmellah | 2019-03-27 08 :22 :50 +0100 (mer. 27 mars 2019) | 1 ligne

ajout de widget à l'IHM

r35 | tvaira | 2019-03-24 09 :00 :56 +0100 (dim. 24 mars 2019) | 1 ligne

Ajout de la version 0.2 de la base de données : intégration de TTN

r34 | tvaira | 2019-03-24 08 :09 :08 +0100 (dim. 24 mars 2019) | 2 lignes

Intégration de la classe [BaseDeDonnees](#) (MySQL) fournie Ajout du bouton pour le paramétrage (à faire)

r33 | erossi1 | 2019-03-22 12 :07 :53 +0100 (ven. 22 mars 2019) | 1 ligne

ajout d'une methode extrairedonnees + extraire port dans [ruche.cpp](#)

r32 | fmellah | 2019-03-22 08 :20 :50 +0100 (ven. 22 mars 2019) | 1 ligne

Commencement del'IHM

r31 | synapse | 2019-03-21 16 :22 :03 +0100 (jeu. 21 mars 2019) | 1 ligne

IHM Terminée

r30 | erossi1 | 2019-03-21 16 :16 :04 +0100 (jeu. 21 mars 2019) | 1 ligne

ajout d'une methode privée dans [ruche.cpp](#) extraire donnees brute

r29 | synapse | 2019-03-21 14 :53 :14 +0100 (jeu. 21 mars 2019) | 1 ligne

Finalisation de la page de details de ruche.

r28 | erossi1 | 2019-03-21 10 :44 :14 +0100 (jeu. 21 mars 2019) | 1 ligne

ajout d'une conversion de la reçu par le protocole mqtt de la base 64 en base 16

r27 | synapse | 2019-03-20 16 :47 :55 +0100 (mer. 20 mars 2019) | 1 ligne

Ajout du projet HoneyBee (Android)

r26 | erossi1 | 2019-03-20 12 :45 :20 +0100 (mer. 20 mars 2019) | 1 ligne

modification slot recevoirMessage dans [ruche.cpp](#)

r25 | erossi1 | 2019-03-20 11 :24 :12 +0100 (mer. 20 mars 2019) | 1 ligne

reagencement du code

r24 | tvaira | 2019-03-18 17 :32 :27 +0100 (lun. 18 mars 2019) | 1 ligne

Suppression erreur fichier baseDeDonnees.cpp

r23 | tvaira | 2019-03-18 17 :31 :30 +0100 (lun. 18 mars 2019) | 1 ligne

Ajout fichier manquant [baseDeDonnees.cpp](#)

r22 | tvaira | 2019-03-17 07 :11 :01 +0100 (dim. 17 mars 2019) | 1 ligne

Modifications pour Doxygen

r21 | tvaira | 2019-03-16 17 :58 :21 +0100 (sam. 16 mars 2019) | 2 lignes

Mise à jour de la classe [BaseDeDonnees](#)

r20 | tvaira | 2019-03-16 09 :07 :06 +0100 (sam. 16 mars 2019) | 2 lignes

Réorganisation du code source

r19 | erossi1 | 2019-03-15 11 :54 :09 +0100 (ven. 15 mars 2019) | 1 ligne

ajout d'avertissement de connexion

r18 | erossi1 | 2019-03-15 11 :45 :10 +0100 (ven. 15 mars 2019) | 1 ligne

continuation foctionalite de communication avec le broker mqtt



r17 | fmellah | 2019-03-15 11 :00 :34 +0100 (ven. 15 mars 2019) | 1 ligne

r16 | tvaira | 2019-03-15 06 :15 :07 +0100 (ven. 15 mars 2019) | 1 ligne

Modification Doxyfile

r15 | erossi1 | 2019-03-14 15 :36 :06 +0100 (jeu. 14 mars 2019) | 1 ligne

ajout de l'attribut horodatage + ses mutateurs

r14 | fmellah | 2019-03-14 15 :24 :15 +0100 (jeu. 14 mars 2019) | 1 ligne

ajout de la connection à la BDD

r13 | erossi1 | 2019-03-14 11 :30 :53 +0100 (jeu. 14 mars 2019) | 1 ligne

modification du fichier [baseDeDonnees.h](#) [baseDeDonnees.cpp](#)

r12 | fmellah | 2019-03-14 11 :26 :21 +0100 (jeu. 14 mars 2019) | 1 ligne

ajout de la classe BDD

r11 | erossi1 | 2019-03-14 11 :18 :41 +0100 (jeu. 14 mars 2019) | 1 ligne

ajout de signaux et de slots et de connect dans les classes [Ruchelhm](#) et [Ruche](#)

r10 | erossi1 | 2019-03-13 16 :56 :48 +0100 (mer. 13 mars 2019) | 1 ligne

declaration des signaux et slot dans la ruche et definition de ceux ci + implemenatation d'un documentation plus pousser pour chaque fichier + mise en place de la classe communicaion avec l'integration de qTmqtt

r9 | tvaira | 2019-03-09 07 :52 :26 +0100 (sam. 09 mars 2019) | 1 ligne

Initialisation Doxygen

r8 | erossi1 | 2019-03-07 16 :29 :22 +0100 (jeu. 07 mars 2019) | 1 ligne

ajout des relation entre les classe

r7 | erossi1 | 2019-02-28 15 :23 :25 +0100 (jeu. 28 févr. 2019) | 1 ligne

ajout de la classe communication + de l'attribut horodatage sur les classe infos

r6 | erossi1 | 2019-02-28 15 :03 :03 +0100 (jeu. 28 févr. 2019) | 1 ligne

ajout des squelettes ddes classe precedement ajouté + ajout des mutateurs

r5 | erossi1 | 2019-02-27 16 :05 :45 +0100 (mer. 27 févr. 2019) | 1 ligne

ajout supplementaire au squelette de la classe [InfosTemperature](#)

r4 | erossi1 | 2019-02-27 15 :55 :42 +0100 (mer. 27 févr. 2019) | 1 ligne

ajout des fichiers [infosTemperature.h](#) [infosHumidite.h](#) [infosPressionAtmospherique.h](#) [infosensoillement.h](#) [infosTemperature.cpp](#) [infosHumidite.cpp](#) [infosPressionAtmospherique.cpp](#) [infosensoillement.cpp](#) + squelette de la classe [InfosTemperature](#)

r3 | erossi1 | 2019-02-27 15 :24 :39 +0100 (mer. 27 févr. 2019) | 1 ligne

ajout de la classe ruche

r2 | erossi1 | 2019-02-27 15 :18 :57 +0100 (mer. 27 févr. 2019) | 1 ligne

ajout de ruchelhm

r1 | www-data | 2019-02-06 20 :09 :43 +0100 (mer. 06 févr. 2019) | 1 ligne

Creating initial repository structure

## 3 Installation

### Fabrication

```
$ qmake
$ make
$ sudo make install
```

### Base de données

#### Serveur MySQL

```
$ sudo apt install mysql-server
```

#### Ruche

```
$ mysql -uroot -ppassword -hlocalhost < ruche-v0.4.sql
```

#### Compte apiculteur

```
$ mysql -uroot -ppassword -hlocalhost
mysql> CREATE USER 'apiculteur'@'%' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON `ruche`.* TO 'apiculteur'@'%';
mysql> FLUSH PRIVILEGES;
```

#### Accès distant

```
$ sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
bind-address 0.0.0.0
```

#### Redémarrage

```
$ sudo systemctl restart mysql.service
```

#### Vérification

```
$ sudo systemctl status mysql.service
```

## 4 README

Nom : Bee-Honey't

Numéro de version : 1.0

#### Auteurs

Clément LAURAIN [laurain.clement.contact@gmail.com](mailto:laurain.clement.contact@gmail.com)

Florentin MELLAH [florentinmellah@gmail.com](mailto:florentinmellah@gmail.com)

Enzo ROSSI [enzo.rossi13160contact@gmail.com](mailto:enzo.rossi13160contact@gmail.com)

## Description

Système autonome permettant de connaître à distance certains paramètres d'une ruche afin d'assurer son suivi et d'évaluer la santé des abeilles.

## Dépôt SVN

<https://svn.riouxsvn.com/bee-honey-t>

## Recette IR

- Étudiant : Clément LAURAIN
- Étudiant : Florentin MELLAH
- Étudiant : Enzo ROSSI

## Base de données MySQL

```
--
-- Base de données: `ruches`
--

DROP DATABASE IF EXISTS `ruches`;

CREATE DATABASE IF NOT EXISTS `ruches`;

USE `ruches`;

--
-- Structure de la table `Apiculteur`
--

CREATE TABLE IF NOT EXISTS `Apiculteur` (
  `idApiculteur` int(11) NOT NULL AUTO_INCREMENT,
  `Nom` varchar(64) NOT NULL,
  `Prenom` varchar(64) NOT NULL,
  `Identifiant` varchar(255) DEFAULT NULL,
  `MotDePasse` varchar(255) DEFAULT NULL,
  `Email` varchar(64) NOT NULL,
  PRIMARY KEY (`idApiculteur`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `Apiculteur` (`Nom`,`Prenom`,`Identifiant`,`MotDePasse`,`Email`)
VALUES('Vaira','Thierry','tvaira','5f4dcc3b5aa765d61d8327deb882cf99','tvaira@free.fr');
INSERT INTO `Apiculteur` (`Nom`,`Prenom`,`Identifiant`,`MotDePasse`,`Email`)
VALUES('Soulie','Jean','jsoulie','','');

--
-- Structure de la table `TTN`
--

CREATE TABLE IF NOT EXISTS `TTN` (
  `idTTN` int(11) NOT NULL AUTO_INCREMENT,
  `idApiculteur` int(11) NOT NULL,
  `Hostname` varchar(64) NOT NULL,
  `Port` int(11) NOT NULL,
  `Username` varchar(64) NOT NULL,
  `Password` varchar(255) DEFAULT NULL,
  `ApplicationID` varchar(64) NOT NULL,
  `Identifiant` varchar(255) DEFAULT NULL,
  `MotDePasse` varchar(255) DEFAULT NULL,
  `Email` varchar(64) NOT NULL,
  PRIMARY KEY (`idTTN`),
  CONSTRAINT TTN_fk_1 FOREIGN KEY (`idApiculteur`) REFERENCES Apiculteur (`idApiculteur`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `TTN` (`idApiculteur`,`Hostname`,`Port`,`Username`,`Password`,`ApplicationID`,`Identifiant`,`MotDePasse`,`Email`)
VALUES('2','eu.thethings.network',1883,'mes_ruches','ttn-account-v2.vC-aqMRnLLzGkNjODWgy81kLqzxBPAT8_mE-L7U2C_w','mes_ruches',
```

```
--
-- Structure de la table 'Ruche'
--

CREATE TABLE IF NOT EXISTS 'Ruche' (
  'idRuche' int(11) NOT NULL AUTO_INCREMENT,
  'idTTN' int(11) NOT NULL,
  'Nom' varchar(255) NOT NULL,
  'Description' varchar(255) DEFAULT NULL,
  'DateMiseEnService' date NOT NULL,
  'Adresse' varchar(255) DEFAULT NULL,
  'Longitude' varchar(15) DEFAULT NULL,
  'Latitude' varchar(15) DEFAULT NULL,
  'DeviceID' varchar(64) NOT NULL,
  PRIMARY KEY ('idRuche'),
  UNIQUE ('DeviceID'),
  CONSTRAINT Ruche_fk_1 FOREIGN KEY ('idTTN') REFERENCES TTN('idTTN') ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO 'Ruche'
  ('idTTN', 'Nom', 'Description', 'DateMiseEnService', 'Adresse', 'Longitude', 'Latitude', 'DeviceID') VALUES('1', 'Ruche 1', '...', '2015-

-- -----

--
-- Structure de la table 'Seuils'
--

CREATE TABLE IF NOT EXISTS 'Seuils' (
  'idSeuils' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'TemperatureIntMin' double NOT NULL,
  'TemperatureIntMax' double NOT NULL,
  'HumiditeIntMin' double NOT NULL,
  'HumiditeIntMax' double NOT NULL,
  'TemperatureExtMin' double NOT NULL,
  'TemperatureExtMax' double NOT NULL,
  'HumiditeExtMin' double NOT NULL,
  'HumiditeExtMax' double NOT NULL,
  'PressionMin' int(11) NOT NULL,
  'PressionMax' int(11) NOT NULL,
  'PoidsMin' int(11) NOT NULL,
  'PoidsMax' int(11) NOT NULL,
  'EnsoleillementMin' int(11) NOT NULL,
  'EnsoleillementMax' int(11) NOT NULL,
  'Charge' int(11) NOT NULL,
  PRIMARY KEY ('idSeuils'),
  CONSTRAINT Seuils_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO 'Seuils' ('idRuche', 'TemperatureIntMin', 'TemperatureIntMax', 'HumiditeIntMin',
  'HumiditeIntMax', 'TemperatureExtMin', 'TemperatureExtMax', 'HumiditeExtMin', 'HumiditeExtMax', 'PressionMin',
  'PressionMax', 'PoidsMin', 'PoidsMax', 'EnsoleillementMin', 'EnsoleillementMax', 'Charge')
  VALUES('1', '25', '35', '20', '30', '5', '35', '20', '35', '1010', '1110', '10', '100', '10', '1000', '25');

-- -----

--
-- Structure de la table 'MesuresEnvironnement'
--

CREATE TABLE IF NOT EXISTS 'MesuresEnvironnement' (
  'idMesuresEnvironnement' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Temperature' double NOT NULL,
  'Humidite' double NOT NULL,
  'Pression' int(11) NOT NULL,
  'Horodatage' datetime NOT NULL,
  PRIMARY KEY ('idMesuresEnvironnement'),
  CONSTRAINT MesuresEnvironnement_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE
    CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----

--
-- Structure de la table 'MesuresEnsoleillement'
--

CREATE TABLE IF NOT EXISTS 'MesuresEnsoleillement' (
  'idMesuresEnsoleillement' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Ensoleillement' int(11) NOT NULL,
  'Horodatage' datetime NOT NULL,
  PRIMARY KEY ('idMesuresEnsoleillement'),
  CONSTRAINT MesuresEnsoleillement_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE
    CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
```

```
--
-- Structure de la table 'MesuresRuche'
--

CREATE TABLE IF NOT EXISTS 'MesuresRuche' (
  'idMesuresRuche' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Temperature' double NOT NULL,
  'Humidite' double NOT NULL,
  'Horodatage' datetime NOT NULL,
  PRIMARY KEY ('idMesuresRuche'),
  CONSTRAINT MesuresRuche_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
--
-- Structure de la table 'MesuresPoids'
--

CREATE TABLE IF NOT EXISTS 'MesuresPoids' (
  'idMesuresPoids' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Poids' int(11) NOT NULL,
  'Horodatage' datetime NOT NULL,
  PRIMARY KEY ('idMesuresPoids'),
  CONSTRAINT MesuresPoids_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
--
-- Structure de la table 'MesuresEnergie'
--

CREATE TABLE IF NOT EXISTS 'MesuresEnergie' (
  'idMesuresEnergie' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Charge' int(11) NOT NULL,
  'Tension' double NOT NULL,
  'Courant' double NOT NULL,
  'Horodatage' datetime NOT NULL,
  PRIMARY KEY ('idMesuresEnergie'),
  CONSTRAINT MesuresEnergie_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
--
-- Structure de la table 'TypeAlerte'
--

CREATE TABLE IF NOT EXISTS 'TypeAlerte' (
  'idType' int(11) NOT NULL AUTO_INCREMENT,
  'Nom' varchar(64) NOT NULL,
  PRIMARY KEY ('idType')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
--
-- Structure de la table 'Alertes'
--

CREATE TABLE IF NOT EXISTS 'Alertes' (
  'idAlerte' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'idType' int(11) NOT NULL,
  'Description' text CHARACTER SET utf8 NOT NULL,
  'Horodatage' datetime NOT NULL,
  PRIMARY KEY ('idAlerte'),
  CONSTRAINT Alertes_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE CASCADE,
  CONSTRAINT Alertes_fk_2 FOREIGN KEY ('idType') REFERENCES TypeAlerte('idType') ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Contenu de la table 'Alertes'
--

--
```

```
-- Structure de la table 'MesuresJournalieresEnvironnement'
--

CREATE TABLE IF NOT EXISTS 'MesuresJournalieresEnvironnement' (
  'idMesuresJournalieresEnvironnement' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Temperature' double NOT NULL,
  'TemperatureMin' double NOT NULL,
  'TemperatureMax' double NOT NULL,
  'Humidite' double NOT NULL,
  'HumiditeMin' double NOT NULL,
  'HumiditeMax' double NOT NULL,
  'Pression' int(11) NOT NULL,
  'PressionMin' int(11) NOT NULL,
  'PressionMax' int(11) NOT NULL,
  'DateMesure' date NOT NULL,
  'HeureMesure' time NOT NULL,
  PRIMARY KEY ('idMesuresJournalieresEnvironnement'),
  CONSTRAINT MesuresJournalieresEnvironnement_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON
    DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----

--
-- Structure de la table 'MesuresJournalieresEnsoleillement'
--

CREATE TABLE IF NOT EXISTS 'MesuresJournalieresEnsoleillement' (
  'idMesuresJournalieresEnsoleillement' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Ensoleillement' int(11) NOT NULL,
  'EnsoleillementMin' int(11) NOT NULL,
  'EnsoleillementMax' int(11) NOT NULL,
  'DateMesure' date NOT NULL,
  'HeureMesure' time NOT NULL,
  PRIMARY KEY ('idMesuresJournalieresEnsoleillement'),
  CONSTRAINT MesuresJournalieresEnsoleillement_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON
    DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----

--
-- Structure de la table 'MesuresJournalieresRuche'
--

CREATE TABLE IF NOT EXISTS 'MesuresJournalieresRuche' (
  'idMesuresJournalieresRuche' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Temperature' double NOT NULL,
  'TemperatureMin' double NOT NULL,
  'TemperatureMax' double NOT NULL,
  'Humidite' double NOT NULL,
  'HumiditeMin' double NOT NULL,
  'HumiditeMax' double NOT NULL,
  'DateMesure' date NOT NULL,
  'HeureMesure' time NOT NULL,
  PRIMARY KEY ('idMesuresJournalieresRuche'),
  CONSTRAINT MesuresJournalieresRuche_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE
    CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----

--
-- Structure de la table 'MesuresJournalieresPoids'
--

CREATE TABLE IF NOT EXISTS 'MesuresJournalieresPoids' (
  'idMesuresJournalieresPoids' int(11) NOT NULL AUTO_INCREMENT,
  'idRuche' int(11) NOT NULL,
  'Poids' int(11) NOT NULL,
  'PoidsMin' int(11) NOT NULL,
  'PoidsMax' int(11) NOT NULL,
  'DateMesure' date NOT NULL,
  'HeureMesure' time NOT NULL,
  PRIMARY KEY ('idMesuresJournalieresPoids'),
  CONSTRAINT MesuresJournalieresPoids_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE
    CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----

--
-- Structure de la table 'MesuresJournalieresEnergie'
--

CREATE TABLE IF NOT EXISTS 'MesuresJournalieresEnergie' (
  'idMesuresJournalieresEnergie' int(11) NOT NULL AUTO_INCREMENT,
```

```

'idRuche' int(11) NOT NULL,
'Charge' int(11) NOT NULL,
'Tension' double NOT NULL,
'Courant' double NOT NULL,
'DateMesure' date NOT NULL,
'HeureMesure' time NOT NULL,
PRIMARY KEY ('idMesuresJournalieresEnergie'),
CONSTRAINT MesuresJournalieresEnergie_fk_1 FOREIGN KEY ('idRuche') REFERENCES Ruche('idRuche') ON DELETE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

## 5 A propos

### Auteur

*Clément* LAURAIN [laurain.clement.contact@gmail.com](mailto:laurain.clement.contact@gmail.com)  
*Florentin* MELLAH [florentinmellah@gmail.com](mailto:florentinmellah@gmail.com)  
*Enzo* ROSSI [enzo.rossi13160contact@gmail.com](mailto:enzo.rossi13160contact@gmail.com)

### Version

1.0

### Date

2019

## 6 Licence GPL

This program is free software ; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation ; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY ; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program ; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

## 7 Liste des choses à faire

### Membre [Communication](#) : [:erreurClientTTN](#) ()

Faire une boîte de dialogue d'information sur l'erreur rencontrée

### Classe [fr.campus.laurainc.honeybee.DetailsRucheActivity](#)

Renommer cette classe

### Membre [fr.campus.laurainc.honeybee.MainActivity.onActivityResult](#) (int requestCode, int resultCode, Intent intent)

Gestion d'autres paramètres ? MQTT ? ...

### Membre [fr.campus.laurainc.honeybee.MainActivity.onCreate](#) (Bundle savedInstanceState)

Affichage de l'état de connexion MySQL ?

### Classe [fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity](#)

D'autres paramètres ? MQTT ? ...

### Membre [Ruchelhm](#) : [:recevoirEtatClientConnexion](#) (bool etat)

récupérer la longitude et la latitude depuis le format Json et afficher la position sur la carte

## 8 Documentation des espaces de nommage

### 8.1 Paquetage fr

#### Paquetages

- package [campus](#)

### 8.2 Paquetage fr.campus

#### Paquetages

- package [laurainc](#)

### 8.3 Paquetage fr.campus.laurainc

#### Paquetages

- package [honeybee](#)

### 8.4 Paquetage fr.campus.laurainc.honeybee

#### Classes

- class [alertesActivity](#)
- class [BaseDeDonnees](#)
  - Gestion d'une base de données MySQL (Singleton)*
- class [CarteActivity](#)
- class [ClientMQTT](#)
- class [DashboardActivity](#)
- class [DetailsRucheActivity](#)
- class [ExampleInstrumentedTest](#)
- class [ExampleUnitTest](#)
- class [GraphActivity](#)
- class [homeActivity](#)
- class [HoneyBee](#)
  - Paramètres globaux de l'application.*
- class [MainActivity](#)
  - Activité principale de l'application (Thread UI)*
- class [NouvelleRucheActivity](#)
- class [ParametresHoneyBeeActivity](#)
  - Activité de paramétrage de l'application.*
- class [Ruche](#)
- class [TimeAsXAxisLabelFormatter](#)



## 8.5 Référence de l'espace de nommage Ui

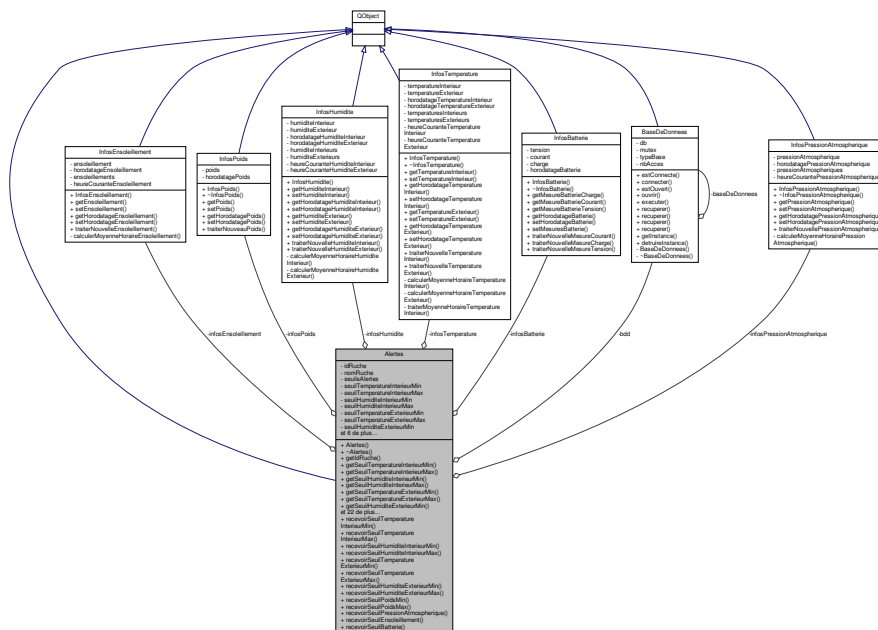
## 9 Documentation des classes

## 9.1 Référence de la classe Alertes

La classe des alertes.

```
#include <alertes.h>
```

Graphe de collaboration de Alertes :



## Connecteurs publics

- void **recevoirSeuilTemperatureInterieurMin** (QString seuil)  
*slot de reception des seuils de temperature interieur minimum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilTemperatureInterieurMax** (QString seuil)  
*slot de reception des seuils de temperature interieur maximum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilHumiditeInterieurMin** (QString seuil)  
*slot de reception des seuils de humidite interieur minimum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilHumiditeInterieurMax** (QString seuil)  
*slot de reception des seuils de humidite interieur maximum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilTemperatureExterieurMin** (QString seuil)  
*slot de reception des seuils de temperature exterieur minimum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilTemperatureExterieurMax** (QString seuil)  
*slot de reception des seuils de temperature exterieur maximum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilHumiditeExterieurMin** (QString seuil)  
*slot de reception des seuils de humidite exterieur minimum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilHumiditeExterieurMax** (QString seuil)  
*slot de reception des seuils de humidite exterieur maximum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilPoidsMin** (QString seuil)  
*slot de reception des seuils de poids minimum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilPoidsMax** (QString seuil)  
*slot de reception des seuils de poids maximum venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilPressionAtmospherique** (QString seuil)  
*slot de reception des seuils de pression atmospherique venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilEnsoleillement** (QString seuil)  
*slot de reception des seuils d'ensoleillement venant de la classe [ReglagesAlertesIhm](#)*
- void **recevoirSeuilBatterie** (QString seuil)

## Signaux

- void `envoiAlertesTemperatureInterieur` (`SeuilsAlertes`)
- void `envoiAlertesTemperatureExterieur` (`SeuilsAlertes`)
- void `envoiAlertesHumiditeInterieur` (`SeuilsAlertes`)
- void `envoiAlertesHumiditeExterieur` (`SeuilsAlertes`)
- void `envoiAlertesPressionAtmospherique` (`SeuilsAlertes`)
- void `envoiAlertesPoids` (`SeuilsAlertes`)
- void `envoiAlertesEnsoleillement` (`SeuilsAlertes`, double)
- void `envoiAlertesBatterie` (`SeuilsAlertes`, double)

## Fonctions membres publiques

- `Alertes` (`QString idRuche`, `QString nomRuche`, `QObject *parent=0`)  
*Constructeur de la classe `Alertes`.*
- `~Alertes` ()
- `QString getIdRuche` ()
- double `getSeuilTemperatureInterieurMin` ()
- double `getSeuilTemperatureInterieurMax` ()
- double `getSeuilHumiditeInterieurMin` ()
- double `getSeuilHumiditeInterieurMax` ()
- double `getSeuilTemperatureExterieurMin` ()
- double `getSeuilTemperatureExterieurMax` ()
- double `getSeuilHumiditeExterieurMin` ()
- double `getSeuilHumiditeExterieurMax` ()
- double `getSeuilPoidsMin` ()
- double `getSeuilPoidsMax` ()
- double `getSeuilPressionAtmospherique` ()
- double `getSeuilEnsoleillement` ()
- double `getSeuilBatterie` ()
- void `setInfosTemperature` (`InfosTemperature *infosTemperature`)
- void `setInfosHumidite` (`InfosHumidite *infosHumidite`)
- void `setInfosPressionAtmospherique` (`InfosPressionAtmospherique *infosPressionAtmospherique`)
- void `setInfosPoids` (`InfosPoids *infosPoids`)
- void `setInfosBatterie` (`InfosBatterie *infosBatterie`)
- void `setInfosEnsoleillement` (`InfosEnsoleillement *infosEnsoleillement`)
- void `alertesTemperatureInterieur` ()  
*defini les seuils d'alertes de la temperature interieur*
- void `alertesTemperatureExterieur` ()  
*defini les seuils d'alertes de la temperature exterieur*
- void `alertesHumiditeInterieur` ()  
*defini les seuils d'alertes de l'humidite interieur*
- void `alertesHumiditeExterieur` ()  
*defini les seuils d'alertes de l'humidite exterieur*
- void `alertesPressionAtmospherique` ()  
*defini les seuils d'alertes de pression atmospherique*
- void `alertesPoids` ()  
*defini les seuils d'alertes de poids*
- void `alertesEnsoleillement` ()  
*defini les seuils d'alertes d'ensoleillement*
- void `alertesBatterie` ()
- void `appelerLesAlertes` (`TypeAlertes typeAlertes`)  
*defini les différents appels des alertes*
- void `envoyerMailAlerte` (`QString email`, `QString objet`, `QString message`)

## Attributs privés

- `QString idRuche`
- `QString nomRuche`
- `InfosTemperature * infosTemperature`
- `InfosHumidite * infosHumidite`
- `InfosPressionAtmospherique * infosPressionAtmospherique`
- `InfosEnsoleillement * infosEnsoleillement`
- `InfosBatterie * infosBatterie`
- `InfosPoids * infosPoids`
- `SeuilsAlertes seuilsAlertes`
- `BaseDeDonnees * bdd`  
*agrégation de l'objet `BaseDeDonnees`*
- double `seuilTemperatureInterieurMin`
- double `seuilTemperatureInterieurMax`
- double `seuilHumiditeInterieurMin`
- double `seuilHumiditeInterieurMax`

- double [seuilTemperatureExterieurMin](#)
- double [seuilTemperatureExterieurMax](#)
- double [seuilHumiditeExterieurMin](#)
- double [seuilHumiditeExterieurMax](#)
- double [seuilPoidsMin](#)
- double [seuilPoidsMax](#)
- double [seuilEnsoleillement](#)
- double [seuilPressionAtmospherique](#)
- double [seuilBatterie](#)

### 9.1.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

### 9.1.2 Documentation des constructeurs et destructeur

#### 9.1.2.1 Alertes()

```
Alertes::Alertes (
    QString idRuche,
    QString nomRuche,
    QObject * parent = 0 ) [explicit]
```

Définition des attributs de la classe [Alertes](#)

Références [bdd](#), [BDD\\_NOMBASE](#), [BDD\\_PASSWORD](#), [BDD\\_SERVEUR](#), [BDD\\_USERNAME](#), [BaseDeDonnees : :connecter\(\)](#), [BaseDeDonnees : :estConnecte\(\)](#), [BaseDeDonnees : :getInstance\(\)](#), et [BaseDeDonnees : :recuperer\(\)](#).

```
00038                                     : QObject (parent),
    idRuche(idRuche), nomRuche(nomRuche),
    infosTemperature(0), infosHumidite(0),
    infosPressionAtmospherique(0),
00039     seuilTemperatureInterieurMin(
TEMPERATURE_INTERIEUR_SEUIL_MIN),
    seuilTemperatureInterieurMax(
TEMPERATURE_INTERIEUR_SEUIL_MAX),
    seuilHumiditeInterieurMin(HUMIDITE_INTERIEUR_SEUIL_MIN
), seuilHumiditeInterieurMax(
HUMIDITE_INTERIEUR_SEUIL_MAX),
    seuilTemperatureExterieurMin(
TEMPERATURE_EXTERIEUR_SEUIL_MIN),
    seuilTemperatureExterieurMax(
TEMPERATURE_EXTERIEUR_SEUIL_MAX),
    seuilHumiditeExterieurMin(HUMIDITE_EXTERIEUR_SEUIL_MIN
), seuilHumiditeExterieurMax(
HUMIDITE_EXTERIEUR_SEUIL_MAX), seuilPoidsMin(
POIDS_SEUIL_MIN), seuilPoidsMax(POIDS_SEUIL_MAX),
    seuilEnsoleillement(ENSOLEILLEMENT_SEUIL_MIN),
    seuilPressionAtmospherique(
PRESSION_ATMOSPHERIQUE_SEUIL_MIN)
00040
00041 {
00042     bdd = BaseDeDonnees::getInstance();
00043     if(!bdd->estConnecte())
00044         bdd->connecter(BDD_NOMBASE, BDD_USERNAME,
BDD_PASSWORD, BDD_SERVEUR);
00045
00046     QStringList seuils;
00047     QString requete = "SELECT * FROM Seuils WHERE idRuche='" + idRuche + "'";
00048     qDebug() << Q_FUNC_INFO << requete;
00049     bool retour = bdd->recuperer(requete, seuils);
00050     if(retour)
```

```

00051      {
00052          qDebug() << Q_FUNC_INFO << seuils;
00053          /*seuilTemperatureInterieurMin = seuils.at(1).toDouble();
00054          seuilTemperatureInterieurMax = ;
00055          seuilHumiditeInterieurMin = ;
00056          seuilHumiditeInterieurMax = ;
00057          seuilTemperatureExterieurMin = ;
00058          seuilTemperatureExterieurMax = ;
00059          seuilHumiditeExterieurMin = ;
00060          seuilHumiditeExterieurMax = ;
00061          seuilPoidsMin = ;
00062          seuilPoidsMax = ;
00063          seuilEnsoleillement = ;
00064          seuilPressionAtmospherique = ;*/
00065      }
00066 }

```

### 9.1.2.2 ~Alertes()

Alertes::~Alertes ( )

Références [BaseDeDonnees](#) : [destruireInstance\(\)](#).

```

00069 {
00070     BaseDeDonnees::destruireInstance();
00071 }

```

## 9.1.3 Documentation des fonctions membres

### 9.1.3.1 alertesBatterie()

void Alertes::alertesBatterie ( )

Références [bdd](#), [bon](#), [envoiAlertesBatterie\(\)](#), [envoyerMailAlerte\(\)](#), [InfosBatterie](#) : [getMesureBatterieCharge\(\)](#), [infosBatterie](#), [nomRuche](#), [BaseDeDonnees](#) : [recuperer\(\)](#), [seuilBatterie](#), et [tropBas](#).

Référencé par [appelerLesAlertes\(\)](#).

```

00408 {
00409     double mesureBatterie = infosBatterie->getMesureBatterieCharge();
00410     QString requete = "SELECT Email FROM Apiculteur WHERE idApiculteur = '3'";
00411     QString mailApiculteur;
00412     bdd->recuperer(requete, mailApiculteur);
00413
00414     qDebug() << Q_FUNC_INFO << "mesureBatterie" << mesureBatterie << "seuilBatterie" <<
seuilBatterie;
00415
00416     if(mesureBatterie < seuilBatterie)
00417     {
00418         emit envoiAlertesBatterie(tropBas, mesureBatterie);
00419         envoyerMailAlerte(mailApiculteur, "Alerte Batterie", "Bonjour, une alerte a été
déctée dans la " + nomRuche + " : niveau de batterie faible.");
00420     }
00421     else
00422     {
00423         emit envoiAlertesBatterie(bon, mesureBatterie);
00424     }
00425 }

```

## 9.1.3.2 alertesEnsoleillement()

```
void Alertes::alertesEnsoleillement ( )
```

## Renvoie

*tropHaut* est definit dans le type enum SeuilsAlertes et retourne 0  
*tropBas* est definit dans le type enum SeuilsAlertes et retourne 1  
*bon* est definit dans le type enum SeuilsAlertes et retourne 2

Références [bdd](#), [bon](#), [envoiAlertesEnsoleillement\(\)](#), [InfosEnsoleillement](#) : [:getEnsoleillement\(\)](#), [infosEnsoleillement](#), [BaseDeDonnees](#) : [:recuperer\(\)](#), [seuilEnsoleillement](#), et [tropBas](#).

Référencé par [appelerLesAlertes\(\)](#).

```
00387 {
00388     double mesureEnsoleillement = infosEnsoleillement->
    getEnsoleillement();
00389     QString requete = "SELECT Email FROM Apiculteur WHERE idApiculteur = '3'";
00390     QString mailApiculteur;
00391     bdd->recuperer(requete, mailApiculteur);
00392
00393     qDebug() << Q_FUNC_INFO << "mesureEnsoleillement" << mesureEnsoleillement << "seuilEnsoleillement" <<
    seuilEnsoleillement;
00394
00395     if (mesureEnsoleillement < seuilEnsoleillement)
00396     {
00397         qDebug() << Q_FUNC_INFO << "TROP BAS";
00398         emit envoiAlertesEnsoleillement(tropBas, mesureEnsoleillement);
00399     }
00400     else
00401     {
00402         qDebug() << Q_FUNC_INFO << "CORRECT";
00403         emit envoiAlertesEnsoleillement(bon, mesureEnsoleillement);
00404     }
00405 }
```

## 9.1.3.3 alertesHumiditeExterieur()

```
void Alertes::alertesHumiditeExterieur ( )
```

## Renvoie

*tropHaut* est definit dans le type enum SeuilsAlertes et retourne 0  
*tropBas* est definit dans le type enum SeuilsAlertes et retourne 1  
*bon* est definit dans le type enum SeuilsAlertes et retourne 2

Références [bdd](#), [bon](#), [envoiAlertesHumiditeExterieur\(\)](#), [envoyerMailAlerte\(\)](#), [InfosHumidite](#) : [:getHumiditeExterieur\(\)](#), [infosHumidite](#), [nomRuche](#), [BaseDeDonnees](#) : [:recuperer\(\)](#), [seuilHumiditeExterieurMax](#), [seuilHumiditeExterieurMin](#), [tropBas](#), et [tropHaut](#).

Référencé par [appelerLesAlertes\(\)](#).

```
00292 {
00293     double mesureHumiditeExterieur = infosHumidite->
    getHumiditeExterieur();
00294     QString requete = "SELECT Email FROM Apiculteur WHERE idApiculteur = '3'";
00295     QString mailApiculteur;
00296     bdd->recuperer(requete, mailApiculteur);
00297
00298     qDebug() << Q_FUNC_INFO << "mesureHumiditeExterieur" << mesureHumiditeExterieur << "
    seuilHumiditeExterieurMax" << seuilHumiditeExterieurMax << "seuilHumiditeExterieurMin" <<
    seuilHumiditeExterieurMin;
00299
00300     if (mesureHumiditeExterieur > seuilHumiditeExterieurMax)
00301     {
00302         emit envoiAlertesHumiditeExterieur(tropHaut);
00303         envoyerMailAlerte(mailApiculteur, "Alerte Extérieur Intérieur", "Bonjour, une
    alerte a été détectée dans la " + nomRuche + ": humidité Extérieur élevée.");
```

```

00304         qDebug() << Q_FUNC_INFO << "TROP HAUT";
00305     }
00306     else if (mesureHumiditeExterieur < seuilHumiditeExterieurMin)
00307     {
00308         qDebug() << Q_FUNC_INFO << "TROP BAS";
00309         emit envoiAlertesHumiditeExterieur(tropBas);
00310         envoyerMailAlerte(mailApiculteur, "Alerte Extérieur Intérieur", "Bonjour, une
alerte a été détectée dans la " + nomRuche + ": humidité Extérieur basse.");
00311     }
00312     else
00313     {
00314         qDebug() << Q_FUNC_INFO << "CORRECT";
00315         emit envoiAlertesHumiditeExterieur(bon);
00316     }
00317 }

```

#### 9.1.3.4 alertesHumiditeInterieur()

```
void Alertes::alertesHumiditeInterieur ( )
```

##### Renvoi

*tropHaut* est definit dans le type enum SeuilsAlertes et retourne 0

*tropBas* est definit dans le type enum SeuilsAlertes et retourne 1

*bon* est definit dans le type enum SeuilsAlertes et retourne 2

Références [bdd](#), [bon](#), [envoiAlertesHumiditeInterieur\(\)](#), [envoyerMailAlerte\(\)](#), [InfosHumidite : :getHumiditeInterieur\(\)](#), [infosHumidite](#), [nomRuche](#), [BaseDeDonnees : :recuperer\(\)](#), [seuilHumiditeInterieurMax](#), [seuilHumiditeInterieurMin](#), [tropBas](#), et [tropHaut](#).

Référencé par [appelerLesAlertes\(\)](#).

```

00256 {
00257     double mesureHumiditeInterieur = infosHumidite->
getHumiditeInterieur();
00258     QString requete = "SELECT Email FROM Apiculteur WHERE idApiculteur = '3'";
00259     QString mailApiculteur;
00260     bdd->recuperer(requete, mailApiculteur);
00261
00262     qDebug() << Q_FUNC_INFO << "mesureHumiditeInterieur" << mesureHumiditeInterieur << "
seuilHumiditeInterieurMax" << seuilHumiditeInterieurMax << "mesureHumiditeInterieur" <<
mesureHumiditeInterieur;
00263
00264     if (mesureHumiditeInterieur > seuilHumiditeInterieurMax)
00265     {
00266         emit envoiAlertesHumiditeInterieur(tropHaut);
00267         envoyerMailAlerte(mailApiculteur, "Alerte Humidité Intérieur", "Bonjour, une
alerte a été détectée dans la " + nomRuche + ": humidité intérieur élevée.");
00268         qDebug() << Q_FUNC_INFO << "TROP HAUT";
00269     }
00270     else if (mesureHumiditeInterieur < seuilHumiditeInterieurMin)
00271     {
00272         emit envoiAlertesHumiditeInterieur(tropBas);
00273         envoyerMailAlerte(mailApiculteur, "Alerte Humidité Intérieur", "Bonjour, une
alerte a été détectée dans la " + nomRuche + ": humidité intérieur basse.");
00274         qDebug() << Q_FUNC_INFO << "TROP BAS";
00275     }
00276     else
00277     {
00278         emit envoiAlertesHumiditeInterieur(bon);
00279         qDebug() << Q_FUNC_INFO << "CORRECT";
00280     }
00281 }

```

## 9.1.3.5 alertesPoids()

```
void Alertes::alertesPoids ( )
```

## Renvoi

*tropHaut* est definit dans le type enum SeuilsAlertes et retourne 0

*tropBas* est definit dans le type enum SeuilsAlertes et retourne 1

*bon* est definit dans le type enum SeuilsAlertes et retourne 2

Références [bdd](#), [bon](#), [envoiAlertesPoids\(\)](#), [envoyerMailAlerte\(\)](#), [InfosPoids : :getPoids\(\)](#), [infosPoids](#), [nomRuche](#), [BaseDeDonnees : :recuperer\(\)](#), [seuilPoidsMax](#), [seuilPoidsMin](#), [tropBas](#), et [tropHaut](#).

Référencé par [appelerLesAlertes\(\)](#).

```
00356 {
00357     double mesurePoids = infosPoids->getPoids();
00358     QString requete = "SELECT Email FROM Apiculteur WHERE idApiculteur = '3'";
00359     QString mailApiculteur;
00360     bdd->recuperer(requete, mailApiculteur);
00361
00362     qDebug() << Q_FUNC_INFO << "mesurePoids" << mesurePoids << "seuilPoidsMax" <<
seuilPoidsMax << "seuilPoidsMin" << seuilPoidsMin;
00363
00364     if(mesurePoids > seuilPoidsMax)
00365     {
00366         emit envoiAlertesPoids(tropHaut);
00367         envoyerMailAlerte(mailApiculteur, "Alerte Poids", "Bonjour, une alerte a été
détectée dans la " + nomRuche + " : poids élevé.");
00368     }
00369     else if (mesurePoids < seuilPoidsMin)
00370     {
00371         qDebug() << Q_FUNC_INFO;
00372         emit envoiAlertesPoids(tropBas);
00373         envoyerMailAlerte(mailApiculteur, "Alerte Poids", "Bonjour, une alerte a été
détectée dans la " + nomRuche + " : poids faible.");
00374     }
00375     else
00376         emit envoiAlertesPoids(bon);
00377 }
```

## 9.1.3.6 alertesPressionAtmospherique()

```
void Alertes::alertesPressionAtmospherique ( )
```

## Renvoi

*tropHaut* est definit dans le type enum SeuilsAlertes et retourne 0

*tropBas* est definit dans le type enum SeuilsAlertes et retourne 1

*bon* est definit dans le type enum SeuilsAlertes et retourne 2

Références [bdd](#), [bon](#), [envoiAlertesPressionAtmospherique\(\)](#), [envoyerMailAlerte\(\)](#), [InfosPressionAtmospherique : :getPressionAtmospherique\(\)](#), [infosPressionAtmospherique](#), [nomRuche](#), [BaseDeDonnees : :recuperer\(\)](#), [seuilPressionAtmospherique](#), et [tropBas](#).

Référencé par [appelerLesAlertes\(\)](#).

```
00327 {
00328     double mesurePressionAtmospherique = infosPressionAtmospherique->
getPressionAtmospherique();
00329     QString requete = "SELECT Email FROM Apiculteur WHERE idApiculteur = '3'";
00330     QString mailApiculteur;
00331     bdd->recuperer(requete, mailApiculteur);
00332
00333     qDebug() << Q_FUNC_INFO << "mesurePressionAtmospherique" << mesurePressionAtmospherique << "
seuilPressionAtmospherique" << seuilPressionAtmospherique;
00334
00335     if (mesurePressionAtmospherique < seuilPressionAtmospherique)
00336     {
```

```

00337         qDebug() << Q_FUNC_INFO << "TROP BAS";
00338         emit envoiAlertesPressionAtmospherique(
tropBas);
00339         envoyerMailAlerte(mailApiculteur, "Alerte Pression Atmosphérique", "Bonjour, une
alerte a été détectée dans la " + nomRuche + " : pression atmosphérique basse.");
00340     }
00341     else
00342     {
00343         qDebug() << Q_FUNC_INFO << "CORRECT";
00344         emit envoiAlertesPressionAtmospherique(
bon);
00345     }
00346 }

```

### 9.1.3.7 alertesTemperatureExterieur()

```
void Alertes::alertesTemperatureExterieur ( )
```

#### Renvoie

*tropHaut* est défini dans le type enum *SeuilsAlertes* et retourne 0

*tropBas* est défini dans le type enum *SeuilsAlertes* et retourne 1

*bon* est défini dans le type enum *SeuilsAlertes* et retourne 2

Références [bdd](#), [bon](#), [envoiAlertesTemperatureExterieur\(\)](#), [envoyerMailAlerte\(\)](#), [InfosTemperature : :getTemperatureExterieur\(\)](#), [infosTemperature](#), [nomRuche](#), [BaseDeDonnees : :recuperer\(\)](#), [seuilTemperatureExterieurMax](#), [seuilTemperatureExterieurMin](#), [tropBas](#), et [tropHaut](#).

Référencé par [appelerLesAlertes\(\)](#).

```

00220 {
00221     double mesureTemperatureExterieur = infosTemperature->
getTemperatureExterieur();
00222     QString requete = "SELECT Email FROM Apiculteur WHERE idApiculteur = '3'";
00223     QString mailApiculteur;
00224     bdd->recuperer(requete, mailApiculteur);
00225
00226     qDebug() << Q_FUNC_INFO << "mesureTemperatureExterieur" << mesureTemperatureExterieur << "
seuilTemperatureExterieurMin" << seuilTemperatureExterieurMin << "
seuilTemperatureExterieurMax" << seuilTemperatureExterieurMax;
00227
00228     if(mesureTemperatureExterieur > seuilTemperatureExterieurMax)
00229     {
00230         emit envoiAlertesTemperatureExterieur(
tropHaut);
00231         envoyerMailAlerte(mailApiculteur, "Alerte Température Extérieure", "Bonjour, une
alerte a été détectée dans la " + nomRuche + " : température extérieure élevée.");
00232         qDebug() << Q_FUNC_INFO << "TROP HAUT";
00233     }
00234     else if (mesureTemperatureExterieur < seuilTemperatureExterieurMin)
00235     {
00236         qDebug() << Q_FUNC_INFO << "TROP BAS";
00237         emit envoiAlertesTemperatureExterieur(
tropBas);
00238         envoyerMailAlerte(mailApiculteur, "Alerte Température Extérieure", "Bonjour, une
alerte a été détectée dans la " + nomRuche + " : température extérieure basse.");
00239     }
00240     else
00241     {
00242         emit envoiAlertesTemperatureExterieur(
bon);
00243         qDebug() << Q_FUNC_INFO << "CORRECT";
00244     }
00245 }

```



## 9.1.3.8 alertesTemperatureInterieur()

```
void Alertes::alertesTemperatureInterieur ( )
```

## Renvoi

*tropHaut* est definit dans le type enum *SeuilsAlertes* et retourne 0  
*tropBas* est definit dans le type enum *SeuilsAlertes* et retourne 1  
*bon* est definit dans le type enum *SeuilsAlertes* et retourne 2

Références [bdd](#), [bon](#), [envoiAlertesTemperatureInterieur\(\)](#), [envoyerMailAlerte\(\)](#), [InfosTemperature : :getTemperatureInterieur\(\)](#), [infosTemperature](#), [nomRuche](#), [BaseDeDonnees : :recuperer\(\)](#), [seuilTemperatureInterieurMax](#), [seuilTemperatureInterieurMin](#), [tropBas](#), et [tropHaut](#).

Référencé par [appelerLesAlertes\(\)](#).

```
00182 {
00183     qDebug() << Q_FUNC_INFO;
00184
00185     double mesureTemperatureInterieur = infosTemperature->
getTemperatureInterieur();
00186     QString requete = "SELECT Email FROM Apiculteur WHERE idApiculteur = '3'";
00187     QString mailApiculteur;
00188     bdd->recuperer(requete, mailApiculteur);
00189
00190     qDebug() << Q_FUNC_INFO << "mesureTemperatureInterieur" << mesureTemperatureInterieur << "
seuilTemperatureInterieurMin" << seuilTemperatureInterieurMin << "
seuilTemperatureInterieurMax" << seuilTemperatureInterieurMax;
00191
00192     if (mesureTemperatureInterieur > seuilTemperatureInterieurMax)
00193     {
00194         qDebug() << Q_FUNC_INFO << "TROP HAUT";
00195         emit envoiAlertesTemperatureInterieur(
tropHaut);
00196         envoyerMailAlerte(mailApiculteur, "Alerte Température Intérieur", "Bonjour, une
alerte a été détectée dans" + nomRuche + ": température intérieur élevée.");
00197     }
00198     else if (mesureTemperatureInterieur < seuilTemperatureInterieurMin)
00199     {
00200         qDebug() << Q_FUNC_INFO << "TROP BAS";
00201         emit envoiAlertesTemperatureInterieur(
tropBas);
00202         envoyerMailAlerte(mailApiculteur, "Alerte Température Intérieur", "Bonjour, une
alerte a été détectée dans" + nomRuche + ": température intérieur basse.");
00203     }
00204     else
00205     {
00206         qDebug() << Q_FUNC_INFO << "CORRECT";
00207         emit envoiAlertesTemperatureInterieur(
bon);
00208     }
00209 }
```

## 9.1.3.9 appelerLesAlertes()

```
void Alertes::appelerLesAlertes (
    TypeAlertes typeAlertes )
```

## Paramètres

<i>typeAlertes</i>	TypeAlertes
--------------------	-------------

Références [alerteBatterie](#), [alerteEnsoleillement](#), [alerteHumiditeExterieur](#), [alerteHumiditeInterieur](#), [alertePoids](#), [alertePressionAtmospherique](#), [alertesBatterie\(\)](#), [alertesEnsoleillement\(\)](#), [alertesHumiditeExterieur\(\)](#), [alertesHumiditeInterieur\(\)](#), [alertesPoids\(\)](#), [alertesPressionAtmospherique\(\)](#), [alertesTemperatureExterieur\(\)](#), [alertesTemperatureInterieur\(\)](#), [alerteTemperatureExterieur](#), [alerteTemperatureInterieur](#), et [toutesLesAlertes](#).

Référencé par [Ruche : :recevoirEnsoleillementTraite\(\)](#), [Ruche : :recevoirMesureChargeTraite\(\)](#), [Ruche : :recevoirMesureHumiditeExterieurTraite\(\)](#), [Ruche : :recevoirMesureHumiditeInterieurTraite\(\)](#), [Ruche : :recevoirMesurePoidsTraite\(\)](#), [Ruche : :recevoirPressionAtmospheriqueTraite\(\)](#), [Ruche : :recevoirTemperatureExterieurTraite\(\)](#), et [Ruche : :recevoirTemperatureInterieurTraite\(\)](#).

```

00433 {
00434     qDebug() << Q_FUNC_INFO << "typeAlertes" << typeAlertes;
00435
00436     switch(typeAlertes)
00437     {
00438         case alerteTemperatureInterieur :
00439             alertesTemperatureInterieur();
00440             break;
00441         case alerteTemperatureExterieur :
00442             alertesTemperatureExterieur();
00443             break;
00444         case alerteHumiditeInterieur :
00445             alertesHumiditeInterieur();
00446             break;
00447         case alerteHumiditeExterieur :
00448             alertesHumiditeExterieur();
00449             break;
00450         case alertePressionAtmospherique :
00451             alertesPressionAtmospherique();
00452             break;
00453         case alertePoids :
00454             alertesPoids();
00455             break;
00456         case alerteEnsoleillement :
00457             alertesEnsoleillement();
00458             break;
00459         case alerteBatterie :
00460             alertesBatterie();
00461             break;
00462         case toutesLesAlertes:
00463             alertesHumiditeExterieur();
00464             alertesHumiditeInterieur();
00465             alertesPressionAtmospherique();
00466             alertesTemperatureExterieur();
00467             alertesTemperatureInterieur();
00468             alertesEnsoleillement();
00469             alertesPoids();
00470             break;
00471     }
00472 }

```

#### 9.1.3.10 envoiAlertesBatterie

```

void Alertes::envoiAlertesBatterie (
    SeuilsAlertes ,
    double ) [signal]

```

Référencé par [alertesBatterie\(\)](#).

#### 9.1.3.11 envoiAlertesEnsoleillement

```

void Alertes::envoiAlertesEnsoleillement (
    SeuilsAlertes ,
    double ) [signal]

```

Référencé par [alertesEnsoleillement\(\)](#).

#### 9.1.3.12 envoiAlertesHumiditeExterieur

```

void Alertes::envoiAlertesHumiditeExterieur (
    SeuilsAlertes ) [signal]

```

Référencé par [alertesHumiditeExterieur\(\)](#).

**9.1.3.13 envoiAlertesHumiditeInterieur**

```
void Alertes::envoiAlertesHumiditeInterieur (
    SeuilsAlertes ) [signal]
```

Référencé par [alertesHumiditeInterieur\(\)](#).

**9.1.3.14 envoiAlertesPoids**

```
void Alertes::envoiAlertesPoids (
    SeuilsAlertes ) [signal]
```

Référencé par [alertesPoids\(\)](#).

**9.1.3.15 envoiAlertesPressionAtmospherique**

```
void Alertes::envoiAlertesPressionAtmospherique (
    SeuilsAlertes ) [signal]
```

Référencé par [alertesPressionAtmospherique\(\)](#).

**9.1.3.16 envoiAlertesTemperatureExterieur**

```
void Alertes::envoiAlertesTemperatureExterieur (
    SeuilsAlertes ) [signal]
```

Référencé par [alertesTemperatureExterieur\(\)](#).

**9.1.3.17 envoiAlertesTemperatureInterieur**

```
void Alertes::envoiAlertesTemperatureInterieur (
    SeuilsAlertes ) [signal]
```

Référencé par [alertesTemperatureInterieur\(\)](#).

### 9.1.3.18 envoyerMailAlerte()

```
void Alertes::envoyerMailAlerte (
    QString email,
    QString objet,
    QString message )
```

Références [PASSWORD\\_GMAIL](#), [TTN\\_EMAIL](#), et [USER\\_GMAIL](#).

Référencé par [alertesBatterie\(\)](#), [alertesHumiditeExterieur\(\)](#), [alertesHumiditeInterieur\(\)](#), [alertesPoids\(\)](#), [alertesPressionAtmospherique\(\)](#), [alertesTemperatureExterieur\(\)](#), et [alertesTemperatureInterieur\(\)](#).

```
00602 {
00603     Sender smtp(QLatin1String("smtp.gmail.com"), 465, Sender::SslConnection);
00604     smtp.setUser(USER_GMAIL);
00605     smtp.setPassword(PASSWORD_GMAIL);
00606
00607     MimeMessage message;
00608     EmailAddress sender(QLatin1String(TTN_EMAIL), QLatin1String("APP_TITRE"));
00609     message.setSender(sender);
00610     EmailAddress to(email);
00611     message.addTo(to);
00612     message.setSubject(objet);
00613     MimeText text;
00614     text.setText(messageAlerte);
00615     message.addPart(&text);
00616
00617     if (!smtp.sendMail(message))
00618     {
00619         qDebug() << Q_FUNC_INFO << "Erreur" << smtp.lastError();
00620         return;
00621     }
00622
00623     smtp.quit();
00624 }
```

### 9.1.3.19 getIdRuche()

```
QString Alertes::getIdRuche ( )
```

Références [idRuche](#).

Référencé par [ReglagesAlertesIhm : :recevoirReglagesAlertes\(\)](#).

```
00074 {
00075     return idRuche;
00076 }
```

### 9.1.3.20 getSeuilBatterie()

```
double Alertes::getSeuilBatterie ( )
```

Références [seuilBatterie](#).

```
00139 {
00140     return seuilBatterie;
00141 }
```

### 9.1.3.21 getSeuilEnsoleillement()

```
double Alertes::getSeuilEnsoleillement ( )
```

Références [seuilEnsoleillement](#).

Référencé par [RucheIhm : :initialiserMarqueurAlerteEnsoleillement\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00134 {  
00135     return seuilEnsoleillement;  
00136 }
```

### 9.1.3.22 getSeuilHumiditeExterieurMax()

```
double Alertes::getSeuilHumiditeExterieurMax ( )
```

Références [seuilHumiditeExterieurMax](#).

Référencé par [RucheIhm : :initialiserMarqueurAlerteHumiditeExterieur\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00114 {  
00115     return seuilHumiditeExterieurMax;  
00116 }
```

### 9.1.3.23 getSeuilHumiditeExterieurMin()

```
double Alertes::getSeuilHumiditeExterieurMin ( )
```

Références [seuilHumiditeExterieurMin](#).

Référencé par [RucheIhm : :initialiserMarqueurAlerteHumiditeExterieur\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00109 {  
00110     return seuilHumiditeExterieurMin;  
00111 }
```

### 9.1.3.24 getSeuilHumiditeInterieurMax()

```
double Alertes::getSeuilHumiditeInterieurMax ( )
```

Références [seuilHumiditeInterieurMax](#).

Référencé par [RucheIhm : :initialiserMarqueurAlerteHumiditeInterieur\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00094 {  
00095     return seuilHumiditeInterieurMax;  
00096 }
```

#### 9.1.3.25 getSeuilHumiditeInterieurMin()

```
double Alertes::getSeuilHumiditeInterieurMin ( )
```

Références [seuilHumiditeInterieurMin](#).

Référencé par [RucheIhm : :initialiserMarqueurAlerteHumiditeInterieur\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00089 {  
00090     return seuilHumiditeInterieurMin;  
00091 }
```

#### 9.1.3.26 getSeuilPoidsMax()

```
double Alertes::getSeuilPoidsMax ( )
```

Références [seuilPoidsMax](#).

Référencé par [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00124 {  
00125     return seuilPoidsMax;  
00126 }
```

#### 9.1.3.27 getSeuilPoidsMin()

```
double Alertes::getSeuilPoidsMin ( )
```

Références [seuilPoidsMin](#).

Référencé par [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00119 {  
00120     return seuilPoidsMin;  
00121 }
```

#### 9.1.3.28 getSeuilPressionAtmospherique()

```
double Alertes::getSeuilPressionAtmospherique ( )
```

Références [seuilPressionAtmospherique](#).

Référencé par [RucheIhm : :initialiserMarqueurAlertePression\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00129 {  
00130     return seuilPressionAtmospherique;  
00131 }
```

### 9.1.3.29 getSeuilTemperatureExterieurMax()

```
double Alertes::getSeuilTemperatureExterieurMax ( )
```

Références [seuilTemperatureExterieurMax](#).

Référencé par [Ruchelhm : :initialiserMarqueurAlerteTemperatureExterieur\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00104 {  
00105     return seuilTemperatureExterieurMax;  
00106 }
```

### 9.1.3.30 getSeuilTemperatureExterieurMin()

```
double Alertes::getSeuilTemperatureExterieurMin ( )
```

Références [seuilTemperatureExterieurMin](#).

Référencé par [Ruchelhm : :initialiserMarqueurAlerteTemperatureExterieur\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00099 {  
00100     return seuilTemperatureExterieurMin;  
00101 }
```

### 9.1.3.31 getSeuilTemperatureInterieurMax()

```
double Alertes::getSeuilTemperatureInterieurMax ( )
```

Références [seuilTemperatureInterieurMax](#).

Référencé par [Ruchelhm : :InitialiserMarqueurAlerteTemperatureInterieur\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00084 {  
00085     return seuilTemperatureInterieurMax;  
00086 }
```

### 9.1.3.32 getSeuilTemperatureInterieurMin()

```
double Alertes::getSeuilTemperatureInterieurMin ( )
```

Références [seuilTemperatureInterieurMin](#).

Référencé par [Ruchelhm : :InitialiserMarqueurAlerteTemperatureInterieur\(\)](#), et [ReglagesAlertesIhm : :showEvent\(\)](#).

```
00079 {  
00080     return seuilTemperatureInterieurMin;  
00081 }
```

### 9.1.3.33 recevoirSeuilBatterie

```
void Alertes::recevoirSeuilBatterie (  
    QString seuil ) [slot]
```

Références [seuilBatterie](#).

```
00597 {  
00598     seuilBatterie = seuil.toDouble();  
00599 }
```

### 9.1.3.34 recevoirSeuilEnsoleillement

```
void Alertes::recevoirSeuilEnsoleillement (  
    QString seuil ) [slot]
```

**Paramètres**

<i>seuil</i>	
--------------	--

Références [seuilEnsoleillement](#).

```
00592 {  
00593     seuilEnsoleillement = seuil.toDouble();  
00594 }
```

**9.1.3.35 recevoirSeuilHumiditeExterieurMax**

```
void Alertes::recevoirSeuilHumiditeExterieurMax (  
    QString seuil )    [slot]
```

**Paramètres**

<i>seuil</i>	
--------------	--

Références [seuilHumiditeExterieurMax](#).

```
00552 {  
00553     seuilHumiditeExterieurMax = seuil.toDouble();  
00554 }
```

**9.1.3.36 recevoirSeuilHumiditeExterieurMin**

```
void Alertes::recevoirSeuilHumiditeExterieurMin (  
    QString seuil )    [slot]
```

**Paramètres**

<i>seuil</i>	
--------------	--

Références [seuilHumiditeExterieurMin](#).

```
00542 {  
00543     seuilHumiditeExterieurMin = seuil.toDouble();  
00544 }
```

**9.1.3.37 recevoirSeuilHumiditeInterieurMax**

```
void Alertes::recevoirSeuilHumiditeInterieurMax (  
    QString seuil )    [slot]
```



## Paramètres

<i>seuil</i>	
--------------	--

Références [seuilHumiditeInterieurMax](#).

```
00512 {  
00513     seuilHumiditeInterieurMax = seuil.toDouble();  
00514 }
```

## 9.1.3.38 recevoirSeuilHumiditeInterieurMin

```
void Alertes::recevoirSeuilHumiditeInterieurMin (  
    QString seuil ) [slot]
```

## Paramètres

<i>seuil</i>	
--------------	--

Références [seuilHumiditeInterieurMin](#).

```
00501 {  
00502     seuilHumiditeInterieurMin = seuil.toDouble();  
00503     qDebug() << Q_FUNC_INFO << seuilHumiditeInterieurMin;  
00504 }
```

## 9.1.3.39 recevoirSeuilPoidsMax

```
void Alertes::recevoirSeuilPoidsMax (  
    QString seuil ) [slot]
```

## Paramètres

<i>seuil</i>	
--------------	--

Références [seuilPoidsMax](#).

```
00572 {  
00573     seuilPoidsMax = seuil.toDouble();  
00574 }
```

## 9.1.3.40 recevoirSeuilPoidsMin

```
void Alertes::recevoirSeuilPoidsMin (  
    QString seuil ) [slot]
```

**Paramètres**

<i>seuil</i>	
--------------	--

Références [seuilPoidsMin](#).

```
00562 {  
00563     seuilPoidsMin = seuil.toDouble\(\);  
00564 }
```

**9.1.3.41 recevoirSeuilPressionAtmospherique**

```
void Alertes::recevoirSeuilPressionAtmospherique (  
    QString seuil )    [slot]
```

**Paramètres**

<i>seuil</i>	
--------------	--

Références [seuilPressionAtmospherique](#).

```
00582 {  
00583     seuilPressionAtmospherique = seuil.toDouble\(\);  
00584 }
```

**9.1.3.42 recevoirSeuilTemperatureExterieurMax**

```
void Alertes::recevoirSeuilTemperatureExterieurMax (  
    QString seuil )    [slot]
```

**Paramètres**

<i>seuil</i>	
--------------	--

Références [seuilTemperatureExterieurMax](#).

```
00532 {  
00533     seuilTemperatureExterieurMax = seuil.toDouble\(\);  
00534 }
```

**9.1.3.43 recevoirSeuilTemperatureExterieurMin**

```
void Alertes::recevoirSeuilTemperatureExterieurMin (  
    QString seuil )    [slot]
```

## Paramètres

<i>seuil</i>	
--------------	--

Références [seuilTemperatureExterieurMin](#).

```
00522 {  
00523     seuilTemperatureExterieurMin = seuil.toDouble\(\);  
00524 }
```

## 9.1.3.44 recevoirSeuilTemperatureInterieurMax

```
void Alertes::recevoirSeuilTemperatureInterieurMax (  
    QString seuil )    [slot]
```

## Paramètres

<i>seuil</i>	
--------------	--

Références [seuilTemperatureInterieurMax](#).

```
00491 {  
00492     seuilTemperatureInterieurMax = seuil.toDouble\(\);  
00493 }
```

## 9.1.3.45 recevoirSeuilTemperatureInterieurMin

```
void Alertes::recevoirSeuilTemperatureInterieurMin (  
    QString seuil )    [slot]
```

## Paramètres

<i>seuil</i>	
--------------	--

Références [seuilTemperatureInterieurMin](#).

```
00480 {  
00481     seuilTemperatureInterieurMin = seuil.toDouble\(\);  
00482     qDebug() << Q_FUNC_INFO << seuilTemperatureInterieurMin;  
00483 }
```

## 9.1.3.46 setInfosBatterie()

```
void Alertes::setInfosBatterie (  
    InfosBatterie * infosBatterie )
```

Références [infosBatterie](#).

Référencé par [Ruche](#) : [:Ruche\(\)](#).

```
00169 {  
00170     this->infosBatterie = infosBatterie;  
00171 }
```

#### 9.1.3.47 setInfosEnsoleillement()

```
void Alertes::setInfosEnsoleillement (  
    InfosEnsoleillement * infosEnsoleillement )
```

Références [infosEnsoleillement](#).

Référencé par [Ruche : :Ruche\(\)](#).

```
00164 {  
00165     this->infosEnsoleillement = infosEnsoleillement;  
00166 }
```

#### 9.1.3.48 setInfosHumidite()

```
void Alertes::setInfosHumidite (  
    InfosHumidite * infosHumidite )
```

Références [infosHumidite](#).

Référencé par [Ruche : :Ruche\(\)](#).

```
00149 {  
00150     this->infosHumidite = infosHumidite;  
00151 }
```

#### 9.1.3.49 setInfosPoids()

```
void Alertes::setInfosPoids (  
    InfosPoids * infosPoids )
```

Références [infosPoids](#).

Référencé par [Ruche : :Ruche\(\)](#).

```
00159 {  
00160     this->infosPoids = infosPoids;  
00161 }
```

### 9.1.3.50 setInfosPressionAtmospherique()

```
void Alertes::setInfosPressionAtmospherique (
    InfosPressionAtmospherique * infosPressionAtmospherique )
```

Références [infosPressionAtmospherique](#).

Référencé par [Ruche : :Ruche\(\)](#).

```
00154 {
00155     this->infosPressionAtmospherique = infosPressionAtmospherique;
00156 }
```

### 9.1.3.51 setInfosTemperature()

```
void Alertes::setInfosTemperature (
    InfosTemperature * infosTemperature )
```

Références [infosTemperature](#).

Référencé par [Ruche : :Ruche\(\)](#).

```
00144 {
00145     this->infosTemperature = infosTemperature;
00146 }
```

## 9.1.4 Documentation des données membres

### 9.1.4.1 bdd

```
BaseDeDonnees* Alertes::bdd [private]
```

Référencé par [Alertes\(\)](#), [alertesBatterie\(\)](#), [alertesEnsoleillement\(\)](#), [alertesHumiditeExterieur\(\)](#), [alertesHumiditeInterieur\(\)](#), [alertesPoids\(\)](#), [alertesPressionAtmospherique\(\)](#), [alertesTemperatureExterieur\(\)](#), et [alertesTemperatureInterieur\(\)](#).

### 9.1.4.2 idRuche

```
QString Alertes::idRuche [private]
```

Référencé par [getIdRuche\(\)](#).

### 9.1.4.3 infosBatterie

```
InfosBatterie* Alertes::infosBatterie [private]
```

Référencé par [alertesBatterie\(\)](#), et [setInfosBatterie\(\)](#).

#### 9.1.4.4 infosEnsoleillement

`InfosEnsoleillement*` Alertes::infosEnsoleillement [private]

Référencé par [alertesEnsoleillement\(\)](#), et [setInfosEnsoleillement\(\)](#).

#### 9.1.4.5 infosHumidite

`InfosHumidite*` Alertes::infosHumidite [private]

Référencé par [alertesHumiditeExterieur\(\)](#), [alertesHumiditeInterieur\(\)](#), et [setInfosHumidite\(\)](#).

#### 9.1.4.6 infosPoids

`InfosPoids*` Alertes::infosPoids [private]

Référencé par [alertesPoids\(\)](#), et [setInfosPoids\(\)](#).

#### 9.1.4.7 infosPressionAtmospherique

`InfosPressionAtmospherique*` Alertes::infosPressionAtmospherique [private]

Référencé par [alertesPressionAtmospherique\(\)](#), et [setInfosPressionAtmospherique\(\)](#).

#### 9.1.4.8 infosTemperature

`InfosTemperature*` Alertes::infosTemperature [private]

Référencé par [alertesTemperatureExterieur\(\)](#), [alertesTemperatureInterieur\(\)](#), et [setInfosTemperature\(\)](#).

#### 9.1.4.9 nomRuche

`QString` Alertes::nomRuche [private]

Référencé par [alertesBatterie\(\)](#), [alertesHumiditeExterieur\(\)](#), [alertesHumiditeInterieur\(\)](#), [alertesPoids\(\)](#), [alertesPressionAtmospherique\(\)](#), [alertesTemperatureExterieur\(\)](#), et [alertesTemperatureInterieur\(\)](#).

#### 9.1.4.10 seuilBatterie

`double` Alertes::seuilBatterie [private]

Référencé par [alertesBatterie\(\)](#), [getSeuilBatterie\(\)](#), et [recevoirSeuilBatterie\(\)](#).

#### 9.1.4.11 seuilEnsoleillement

```
double Alertes::seuilEnsoleillement [private]
```

Référencé par [alertesEnsoleillement\(\)](#), [getSeuilEnsoleillement\(\)](#), et [recevoirSeuilEnsoleillement\(\)](#).

#### 9.1.4.12 seuilHumiditeExterieurMax

```
double Alertes::seuilHumiditeExterieurMax [private]
```

Référencé par [alertesHumiditeExterieur\(\)](#), [getSeuilHumiditeExterieurMax\(\)](#), et [recevoirSeuilHumiditeExterieurMax\(\)](#).

#### 9.1.4.13 seuilHumiditeExterieurMin

```
double Alertes::seuilHumiditeExterieurMin [private]
```

Référencé par [alertesHumiditeExterieur\(\)](#), [getSeuilHumiditeExterieurMin\(\)](#), et [recevoirSeuilHumiditeExterieurMin\(\)](#).

#### 9.1.4.14 seuilHumiditeInterieurMax

```
double Alertes::seuilHumiditeInterieurMax [private]
```

Référencé par [alertesHumiditeInterieur\(\)](#), [getSeuilHumiditeInterieurMax\(\)](#), et [recevoirSeuilHumiditeInterieurMax\(\)](#).

#### 9.1.4.15 seuilHumiditeInterieurMin

```
double Alertes::seuilHumiditeInterieurMin [private]
```

Référencé par [alertesHumiditeInterieur\(\)](#), [getSeuilHumiditeInterieurMin\(\)](#), et [recevoirSeuilHumiditeInterieurMin\(\)](#).

#### 9.1.4.16 seuilPoidsMax

```
double Alertes::seuilPoidsMax [private]
```

Référencé par [alertesPoids\(\)](#), [getSeuilPoidsMax\(\)](#), et [recevoirSeuilPoidsMax\(\)](#).

#### 9.1.4.17 seuilPoidsMin

```
double Alertes::seuilPoidsMin [private]
```

Référencé par [alertesPoids\(\)](#), [getSeuilPoidsMin\(\)](#), et [recevoirSeuilPoidsMin\(\)](#).

#### 9.1.4.18 seuilPressionAtmospherique

```
double Alertes::seuilPressionAtmospherique [private]
```

Référencé par [alertesPressionAtmospherique\(\)](#), [getSeuilPressionAtmospherique\(\)](#), et [recevoirSeuilPressionAtmospherique\(\)](#).

#### 9.1.4.19 seuilsAlertes

```
SeuilsAlertes Alertes::seuilsAlertes [private]
```

#### 9.1.4.20 seuilTemperatureExterieurMax

```
double Alertes::seuilTemperatureExterieurMax [private]
```

Référencé par [alertesTemperatureExterieur\(\)](#), [getSeuilTemperatureExterieurMax\(\)](#), et [recevoirSeuilTemperatureExterieurMax\(\)](#).

#### 9.1.4.21 seuilTemperatureExterieurMin

```
double Alertes::seuilTemperatureExterieurMin [private]
```

Référencé par [alertesTemperatureExterieur\(\)](#), [getSeuilTemperatureExterieurMin\(\)](#), et [recevoirSeuilTemperatureExterieurMin\(\)](#).

#### 9.1.4.22 seuilTemperatureInterieurMax

```
double Alertes::seuilTemperatureInterieurMax [private]
```

Référencé par [alertesTemperatureInterieur\(\)](#), [getSeuilTemperatureInterieurMax\(\)](#), et [recevoirSeuilTemperatureInterieurMax\(\)](#).

#### 9.1.4.23 seuilTemperatureInterieurMin

```
double Alertes::seuilTemperatureInterieurMin [private]
```

Référencé par [alertesTemperatureInterieur\(\)](#), [getSeuilTemperatureInterieurMin\(\)](#), et [recevoirSeuilTemperatureInterieurMin\(\)](#).

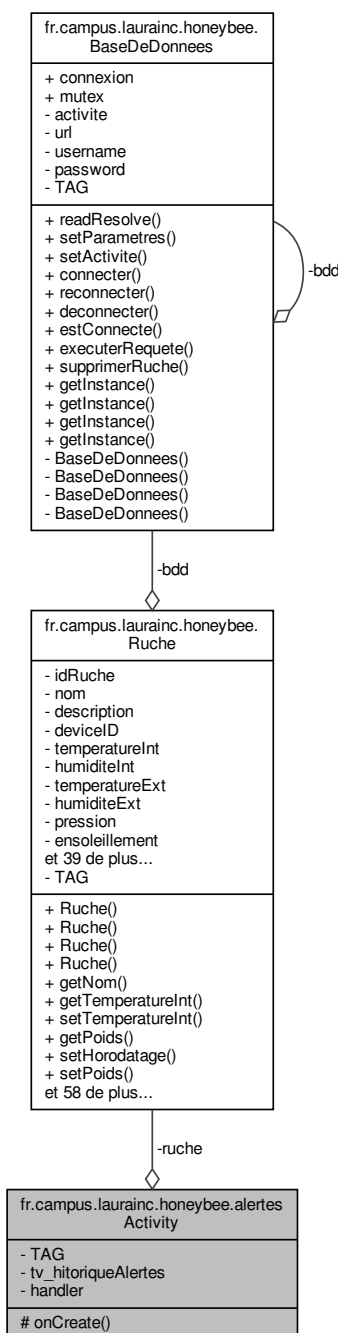
La documentation de cette classe a été générée à partir des fichiers suivants :

- [alertes.h](#)
- [alertes.cpp](#)



## 9.2 Référence de la classe fr.campus.laurainc.honeybee.alertesActivity

Graphe de collaboration de fr.campus.laurainc.honeybee.alertesActivity :



## Fonctions membres protégées

- void `onCreate` (Bundle savedInstanceState)

## Attributs privés

- final String `TAG` = "alertesActivity"
- TextView `tv_historiqueAlertes`
- Ruche `ruche`
- final Handler `handler`

## 9.2.1 Documentation des fonctions membres

### 9.2.1.1 onCreate()

```
void fr.campus.laurainc.honeybee.alertesActivity.onCreate (
    Bundle savedInstanceState ) [protected]
```

Références [fr.campus.laurainc.honeybee.alertesActivity.handler](#), et [fr.campus.laurainc.honeybee.Ruche.recupererHistoriqueAlertes\(\)](#).

```
00017                                     {
00018         super.onCreate(savedInstanceState);
00019         setContentView(R.layout.activity_alertes);
00020
00021         ruche = new Ruche(handler);
00022
00023         tv_historiqueAlertes = findViewById(R.id.tv_historiqueAlertes);
00024         ruche.recupererHistoriqueAlertes();
00025
00026     }
```

## 9.2.2 Documentation des données membres

### 9.2.2.1 handler

```
final Handler fr.campus.laurainc.honeybee.alertesActivity.handler [private]
```

Valeur initiale :

```
= new Handler()
{
    public void handleMessage(Message msg)
    {
        super.handleMessage(msg);
        switch (msg.what)
        {
            case HoneyBee.REQUETE_SQL_ERREUR:
                Log.d(TAG, "handleMessage -> REQUETE SQL ERREUR");
                break;
            case HoneyBee.REQUETE_SQL_ALERTES:
                Log.d(TAG, "handleMessage -> REQUETE SQL ALERTES");
                tv_historiqueAlertes.setText(ruche.
                    getHistoriqueAlertes());
                break;
        }
    }
}
```

Référencé par [fr.campus.laurainc.honeybee.alertesActivity.onCreate\(\)](#).

### 9.2.2.2 ruche

```
Ruche fr.campus.laurainc.honeybee.alertesActivity.ruche [private]
```

## 9.2.2.3 TAG

```
final String fr.campus.laurainc.honeybee.alertesActivity.TAG = "alertesActivity" [private]
```

## 9.2.2.4 tv\_historiqueAlertes

```
TextView fr.campus.laurainc.honeybee.alertesActivity.tv_historiqueAlertes [private]
```

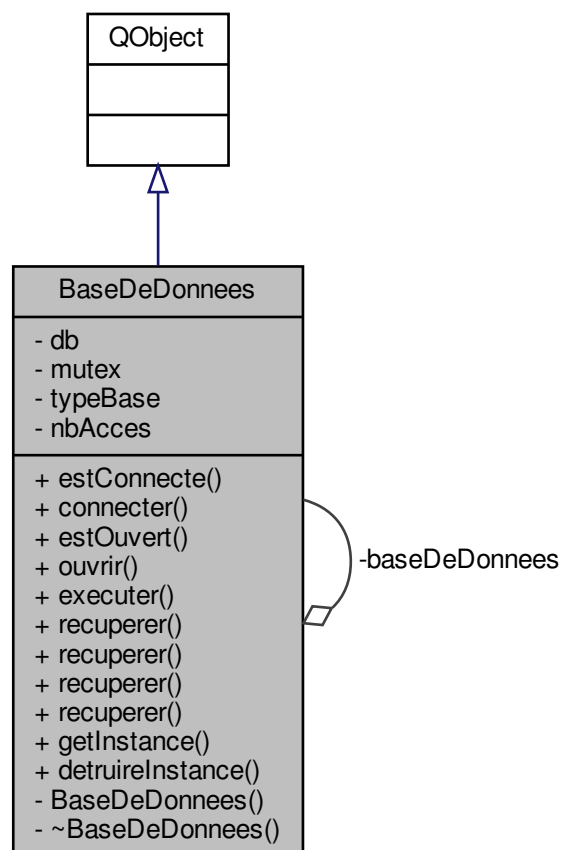
La documentation de cette classe a été générée à partir du fichier suivant :

— [alertesActivity.java](#)

## 9.3 Référence de la classe BaseDeDonnees

```
#include <baseDeDonnees.h>
```

Graphe de collaboration de BaseDeDonnees :



## Fonctions membres publiques

- bool `estConnecte` ()
- bool `connecter` (QString nomBase=`BDD_NOMBASE`, QString username=`BDD_USERNAME`, QString password=`BDD_PAS←`  
`SWORD`, QString serveur=`BDD_SERVEUR`)
- bool `estOuvert` ()
- bool `ouvrir` (QString fichierBase)
- bool `executer` (QString requete)
- bool `recuperer` (QString requete, QString &donnees)
- bool `recuperer` (QString requete, QStringList &donnees)
- bool `recuperer` (QString requete, QVector< QString > &donnees)
- bool `recuperer` (QString requete, QVector< QStringList > &donnees)

## Fonctions membres publiques statiques

- static `BaseDeDonnees` \* `getInstance` (QString type="QMYSQL")
- static void `destruireInstance` ()

## Fonctions membres privées

- `BaseDeDonnees` (QString type)
- `~BaseDeDonnees` ()

## Attributs privés

- QSqlDatabase `db`
- QMutex `mutex`

## Attributs privés statiques

- static `BaseDeDonnees` \* `baseDeDonnees` = NULL
- static QString `typeBase` = "QMYSQL"
- static int `nbAcces` = 0

## 9.3.1 Documentation des constructeurs et destructeur

## 9.3.1.1 BaseDeDonnees()

```
BaseDeDonnees::BaseDeDonnees (
    QString type ) [private]
```

Références `db`, et `typeBase`.

Référencé par `getInstance()`.

```
00023 {
00024     #ifdef DEBUG_BASEDEDONNEES
00025     qDebug() << Q_FUNC_INFO << type;
00026     #endif
00027     db = QSqlDatabase::addDatabase(type);
00028     typeBase = type;
00029 }
```

## 9.3.1.2 ~BaseDeDonnees()

```
BaseDeDonnees::~~BaseDeDonnees ( ) [private]
```

```
00032 {
00033     #ifdef DEBUG_BASEDEDONNEES
00034     qDebug() << Q_FUNC_INFO;
00035     #endif
00036 }
```

## 9.3.2 Documentation des fonctions membres

## 9.3.2.1 connecter()

```
bool BaseDeDonnees::connecter (
    QString nomBase = BDD_NOMBASE,
    QString username = BDD_USERNAME,
    QString password = BDD_PASSWORD,
    QString serveur = BDD_SERVEUR )
```

Références [APP\\_TITRE](#), [db](#), [mutex](#), et [typeBase](#).

Référencé par [Alertes::Alertes\(\)](#), [NouvelleRuchelhm::NouvelleRuchelhm\(\)](#), [ReglagesAlertesIhm::ReglagesAlertesIhm\(\)](#), [Ruche::Ruche\(\)](#), et [Ruchelhm::Ruchelhm\(\)](#).

```
00077 {
00078     if(typeBase != "QMYSQL")
00079         return false;
00080     QMutexLocker verrou(&mutex);
00081     if(!db.isOpen())
00082     {
00083         db.setHostName(serveur);
00084         db.setUserName(username);
00085         db.setPassword(password);
00086         db.setDatabaseName(nomBase);
00087
00088         #ifdef DEBUG_BASEDEDONNEES
00089         qDebug() << Q_FUNC_INFO;
00090         qDebug() << "HostName : " << db.hostName();
00091         qDebug() << "UserName : " << db.userName();
00092         qDebug() << "DatabaseName : " << db.databaseName();
00093         #endif
00094         if(db.open())
00095         {
00096             #ifdef DEBUG_BASEDEDONNEES
00097             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Connexion réussie à %1").arg(
00098 db.hostName());
00099             #endif
00100             return true;
00101         }
00102         else
00103         {
00104             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : impossible de se connecter à la base de
données !");
00105             QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Impossible de
se connecter à la base de données !"));
00106             return false;
00107         }
00108     }
00109     else
00110         return true;
00111 }
```

### 9.3.2.2 detruireInstance()

```
void BaseDeDonnees::detruireInstance ( ) [static]
```

Références [baseDeDonnees](#), et [nbAcces](#).

Référencé par [Alertes : :~Alertes\(\)](#), [NouvelleRuchelhm : :~NouvelleRuchelhm\(\)](#), [ReglagesAlertesIhm : :~ReglagesAlertesIhm\(\)](#), [Ruche : :~Ruche\(\)](#), et [Ruchelhm : :~Ruchelhm\(\)](#).

```
00052 {
00053     // instance ?
00054     if (baseDeDonnees != NULL)
00055     {
00056         if (nbAcces > 0)
00057             nbAcces--;
00058         #ifdef DEBUG_BASEDEDONNEES
00059         qDebug() << Q_FUNC_INFO << "nbAcces restants" << nbAcces;
00060         #endif
00061         // dernier ?
00062         if (nbAcces == 0)
00063         {
00064             delete baseDeDonnees;
00065             baseDeDonnees = NULL;
00066         }
00067     }
00068 }
```

### 9.3.2.3 estConnecte()

```
bool BaseDeDonnees::estConnecte ( )
```

Références [db](#), et [mutex](#).

Référencé par [Alertes : :Alertes\(\)](#), [NouvelleRuchelhm : :NouvelleRuchelhm\(\)](#), [ReglagesAlertesIhm : :ReglagesAlertesIhm\(\)](#), [Ruche : :Ruche\(\)](#), et [Ruchelhm : :Ruchelhm\(\)](#).

```
00071 {
00072     QMutexLocker verrou(&mutex);
00073     return db.isOpen();
00074 }
```

### 9.3.2.4 estOuvert()

```
bool BaseDeDonnees::estOuvert ( )
```

Références [db](#), et [mutex](#).

```
00113 {
00114     QMutexLocker verrou(&mutex);
00115     return db.isOpen();
00116 }
```

## 9.3.2.5 executer()

```
bool BaseDeDonnees::executer (
    QString requete )
```

Références [db](#), et [mutex](#).

Référencé par [Ruchelhm : :afficherAlertesBatterie\(\)](#), [Ruchelhm : :afficherAlertesEnsoleillement\(\)](#), [Ruchelhm : :afficherAlertesHumiditeExterieur\(\)](#), [Ruchelhm : :afficherAlertesHumiditeInterieur\(\)](#), [Ruchelhm : :afficherAlertesPoids\(\)](#), [Ruchelhm : :afficherAlertesPressionAtmospherique\(\)](#), [Ruchelhm : :afficherAlertesTemperatureExterieur\(\)](#), [Ruchelhm : :afficherAlertesTemperatureInterieur\(\)](#), [Ruche : :insererDonneesPortBatterie\(\)](#), [Ruche : :insererDonneesPortEnsoleillement\(\)](#), [Ruche : :insererDonneesPortMesureEnvironnement\(\)](#), [Ruche : :insererDonneesPortMesureRuche\(\)](#), [Ruche : :insererDonneesPortPoids\(\)](#), [Ruche : :insererMesureHoraireEnsoleillement\(\)](#), [Ruche : :insererMesureHoraireEnvironnement\(\)](#), [Ruche : :insererMesureHoraireRuche\(\)](#), [NouvelleRuche : :recevoirDonneeAjoutRuche\(\)](#), [ReglagesAlertesIhm : :recevoirReglagesAlertes\(\)](#), et [Ruchelhm : :supprimerRuche\(\)](#).

```
00151 {
00152     QMutexLocker verrou(&mutex);
00153     QSqlQuery r;
00154     bool retour;
00155
00156     if(db.isOpen())
00157     {
00158         if(requete.contains("UPDATE") || requete.contains("INSERT") || requete.contains("DELETE"))
00159         {
00160             retour = r.exec(requete);
00161             #ifdef DEBUG_BASEDEDONNEES
00162             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Retour %1 pour la requete : %2").arg(
00163                 QString::number(retour)).arg(requete);
00164             #endif
00165             if(retour)
00166             {
00167                 return true;
00168             }
00169             else
00170             {
00171                 qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(r.
00172                     lastError().text()).arg(requete);
00173                 return false;
00174             }
00175             }
00176         else
00177         {
00178             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete);
00179             return false;
00180         }
00181     }
00182     return false;
00183 }
```

## 9.3.2.6 getInstance()

```
BaseDeDonnees * BaseDeDonnees::getInstance (
    QString type = "QMYSQL" ) [static]
```

Références [BaseDeDonnees\(\)](#), [baseDeDonnees](#), et [nbAcces](#).

Référencé par [Alertes : :Alertes\(\)](#), [NouvelleRuchelhm : :NouvelleRuchelhm\(\)](#), [ReglagesAlertesIhm : :ReglagesAlertesIhm\(\)](#), [Ruche : :Ruche\(\)](#), et [Ruchelhm : :Ruchelhm\(\)](#).

```
00039 {
00040     if(baseDeDonnees == NULL)
00041         baseDeDonnees = new BaseDeDonnees(type);
00042
00043     nbAcces++;
00044     #ifdef DEBUG_BASEDEDONNEES
00045     qDebug() << Q_FUNC_INFO << "nbAcces" << nbAcces;
00046     #endif
00047     return baseDeDonnees;
00048 }
00049 }
```

### 9.3.2.7 ouvrir()

```
bool BaseDeDonnees::ouvrir (
    QString fichierBase )
```

Références [db](#), [mutex](#), et [typeBase](#).

```
00119 {
00120     if(typeBase != "SQLITE")
00121         return false;
00122     QMutexLocker verrou(&mutex);
00123     if(!db.isOpen())
00124     {
00125         db.setDatabaseName(fichierBase);
00126
00127         #ifdef DEBUG_BASEDEDONNEES
00128         qDebug() << Q_FUNC_INFO << db.databaseName();
00129         #endif
00130         if(db.open())
00131         {
00132             #ifdef DEBUG_BASEDEDONNEES
00133             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Ouvertir réussie à %1").arg(
00134 db.databaseName());
00135             #endif
00136             return true;
00137         }
00138         else
00139         {
00140             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : impossible d'ouvrir la base de données !"
00141 );
00142             QMessageBox::critical(0, QString::fromUtf8("BaseDeDonnees"), QString::fromUtf8("Impossible
00143 d'ouvrir la base de données !"));
00144             return false;
00145         }
00146     }
00147 }
```

### 9.3.2.8 recuperer() [1/4]

```
bool BaseDeDonnees::recuperer (
    QString requete,
    QString & donnees )
```

Références [db](#), et [mutex](#).

Référencé par [Ruchelhm : :afficherMesuresJournalieresEnsoleillement\(\)](#), [Ruchelhm : :afficherMesuresJournalieresEnvironnement\(\)](#), [Ruchelhm : :afficherMesuresJournalieresRuche\(\)](#), [Ruchelhm : :afficherMesuresJournalieresSelectionnee\(\)](#), [Alertes : :Alertes\(\)](#), [Alertes : :alertesBatterie\(\)](#), [Alertes : :alertesEnsoleillement\(\)](#), [Alertes : :alertesHumiditeExterieur\(\)](#), [Alertes : :alertesHumiditeInterieur\(\)](#), [Alertes : :alertesPoids\(\)](#), [Alertes : :alertesPressionAtmospherique\(\)](#), [Alertes : :alertesTemperatureExterieur\(\)](#), [Alertes : :alertesTemperatureInterieur\(\)](#), [Ruchelhm : :mettreAJourListeRuches\(\)](#), [NouvelleRuchelhm : :NouvelleRuchelhm\(\)](#), et [NouvelleRuchelhm : :recevoirDonneeAjoutRuche\(\)](#).

```
00189 {
00190     QMutexLocker verrou(&mutex);
00191     QSqlQuery r;
00192     bool retour;
00193
00194     if(db.isOpen())
00195     {
00196         if(requete.contains("SELECT"))
00197         {
00198             retour = r.exec(requete);
00199             #ifdef DEBUG_BASEDEDONNEES
00200             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Retour %1 pour la requete : %2").arg(
00201 QString::number(retour)).arg(requete);
00202             #endif
00203             if(retour)
00204             {
00205                 // on se positionne sur l'enregistrement
```



```

00205         r.first();
00206
00207         // on vérifie l'état de l'enregistrement retourné
00208         if(!r.isValid())
00209         {
00210             #ifdef DEBUG_BASEDEDONNEES
00211             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Résultat non valide !");
00212             #endif
00213             return false;
00214         }
00215
00216         // on récupère sous forme de QString la valeur du champ
00217         if(r.isNull(0))
00218         {
00219             #ifdef DEBUG_BASEDEDONNEES
00220             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Aucun résultat !");
00221             #endif
00222             return false;
00223         }
00224         donnees = r.value(0).toString();
00225         #ifdef DEBUG_BASEDEDONNEES
00226         qDebug() << Q_FUNC_INFO << "Enregistrement -> " << donnees;
00227         #endif
00228         return true;
00229     }
00230     else
00231     {
00232         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(r.
lastError().text()).arg(requete);
00233         return false;
00234     }
00235 }
00236 else
00237 {
00238     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete
);
00239     return false;
00240 }
00241 }
00242 else
00243     return false;
00244 }

```

### 9.3.2.9 recuperer() [2/4]

```

bool BaseDeDonnees::recuperer (
    QString requete,
    QStringList & donnees )

```

Références [db](#), et [mutex](#).

```

00250 {
00251     QMutexLocker verrou(&mutex);
00252     QSqlQuery r;
00253     bool retour;
00254
00255     if(db.isOpen())
00256     {
00257         if(requete.contains("SELECT"))
00258         {
00259             retour = r.exec(requete);
00260             #ifdef DEBUG_BASEDEDONNEES
00261             qDebug() << QString::fromUtf8("<BaseDeDonnees::recuperer(QString, QStringList)> retour %1 pour
la requete : %2").arg(QString::number(retour)).arg(requete);
00262             #endif
00263             if(retour)
00264             {
00265                 // on se positionne sur l'enregistrement
00266                 r.first();
00267
00268                 // on vérifie l'état de l'enregistrement retourné
00269                 if(!r.isValid())
00270                 {
00271                     #ifdef DEBUG_BASEDEDONNEES
00272                     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Résultat non valide !");
00273                     #endif
00274                     return false;
00275                 }
00276
00277                 // on récupère sous forme de QString la valeur de tous les champs sélectionnés

```

```

00278         // et on les stocke dans une liste de QString
00279         for(int i=0;i<r.record().count();i++)
00280             if(!r.isNull(i))
00281                 donnees << r.value(i).toString();
00282         #ifdef DEBUG_BASEDEDONNEES
00283         qDebug() << Q_FUNC_INFO << "Enregistrement -> " << donnees;
00284         #endif
00285         return true;
00286     }
00287     else
00288     {
00289         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(r.
lastError().text()).arg(requete);
00290         return false;
00291     }
00292 }
00293 else
00294 {
00295     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete
);
00296     return false;
00297 }
00298 }
00299 else
00300     return false;
00301 }

```

### 9.3.2.10 recuperer() [3/4]

```

bool BaseDeDonnees::recuperer (
    QString requete,
    QVector< QString > & donnees )

```

Références [db](#), et [mutex](#).

```

00307 {
00308     QMutexLocker verrou(&mutex);
00309     QSqlQuery r;
00310     bool retour;
00311     QString data;
00312
00313     if(db.isOpen())
00314     {
00315         if(requete.contains("SELECT"))
00316         {
00317             retour = r.exec(requete);
00318             #ifdef DEBUG_BASEDEDONNEES
00319             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Retour %1 pour la requete : %2").arg(
QString::number(retour)).arg(requete);
00320             #endif
00321             if(retour)
00322             {
00323                 // pour chaque enregistrement
00324                 while ( r.next() )
00325                 {
00326                     // on récupère sous forme de QString la valeur du champs sélectionné
00327                     data = r.value(0).toString();
00328
00329                     #ifdef DEBUG_BASEDEDONNEES
00330                     //qDebug() << Q_FUNC_INFO << "Enregistrement -> " << data;
00331                     #endif
00332
00333                     // on stocke l'enregistrement dans le QVector
00334                     donnees.push_back(data);
00335                 }
00336                 #ifdef DEBUG_BASEDEDONNEES
00337                 qDebug() << Q_FUNC_INFO << "Enregistrement -> " << donnees;
00338                 #endif
00339                 return true;
00340             }
00341             else
00342             {
00343                 qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(r.
lastError().text()).arg(requete);
00344                 return false;
00345             }
00346         }
00347     else
00348     {
00349         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete

```

```

    );
00350         return false;
00351     }
00352 }
00353 else
00354     return false;
00355 }

```

### 9.3.2.11 recuperer() [4/4]

```

bool BaseDeDonnees::recuperer (
    QString requete,
    QVector< QStringList > & donnees )

```

Références [db](#), et [mutex](#).

```

00361 {
00362     QMutexLocker verrou(&mutex);
00363     QSqlQuery r;
00364     bool retour;
00365     QStringList data;
00366
00367     if(db.isOpen())
00368     {
00369         if(requete.contains("SELECT"))
00370         {
00371             retour = r.exec(requete);
00372             #ifdef DEBUG_BASEDEDONNEES
00373             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Retour %1 pour la requete : %2").arg(
00374                 QString::number(retour)).arg(requete);
00375             #endif
00376             if(retour)
00377             {
00378                 // pour chaque enregistrement
00379                 while ( r.next() )
00380                 {
00381                     // on récupère sous forme de QString la valeur de tous les champs sélectionnés
00382                     // et on les stocke dans une liste de QString
00383                     for(int i=0;i<r.record().count();i++)
00384                         data << r.value(i).toString();
00385
00386                     #ifdef DEBUG_BASEDEDONNEES
00387                     //qDebug() << Q_FUNC_INFO << "Enregistrement -> " << data;
00388                     /*for(int i=0;i<r.record().count();i++)
00389                         qDebug() << r.value(i).toString();*/
00390                     #endif
00391
00392                     // on stocke l'enregistrement dans le QVector
00393                     donnees.push_back(data);
00394
00395                     // on efface la liste de QString pour le prochain enregistrement
00396                     data.clear();
00397                 }
00398                 #ifdef DEBUG_BASEDEDONNEES
00399                 qDebug() << Q_FUNC_INFO << "Enregistrement -> " << donnees;
00400                 #endif
00401             }
00402             else
00403             {
00404                 qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(r.
00405                     lastError().text()).arg(requete);
00406                 return false;
00407             }
00408         }
00409         else
00410         {
00411             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete);
00412             return false;
00413         }
00414     }
00415     else
00416         return false;
00417 }

```

### 9.3.3 Documentation des données membres

### 9.3.3.1 baseDeDonnees

```
BaseDeDonnees * BaseDeDonnees::baseDeDonnees = NULL [static], [private]
```

Référencé par [destruireInstance\(\)](#), et [getInstance\(\)](#).

### 9.3.3.2 db

```
QSqlDatabase BaseDeDonnees::db [private]
```

Référencé par [BaseDeDonnees\(\)](#), [connecter\(\)](#), [estConnecte\(\)](#), [estOuvert\(\)](#), [executer\(\)](#), [ouvrir\(\)](#), et [recuperer\(\)](#).

### 9.3.3.3 mutex

```
QMutex BaseDeDonnees::mutex [private]
```

Référencé par [connecter\(\)](#), [estConnecte\(\)](#), [estOuvert\(\)](#), [executer\(\)](#), [ouvrir\(\)](#), et [recuperer\(\)](#).

### 9.3.3.4 nbAcces

```
int BaseDeDonnees::nbAcces = 0 [static], [private]
```

Référencé par [destruireInstance\(\)](#), et [getInstance\(\)](#).

### 9.3.3.5 typeBase

```
QString BaseDeDonnees::typeBase = "QMYSQL" [static], [private]
```

Référencé par [BaseDeDonnees\(\)](#), [connecter\(\)](#), et [ouvrir\(\)](#).

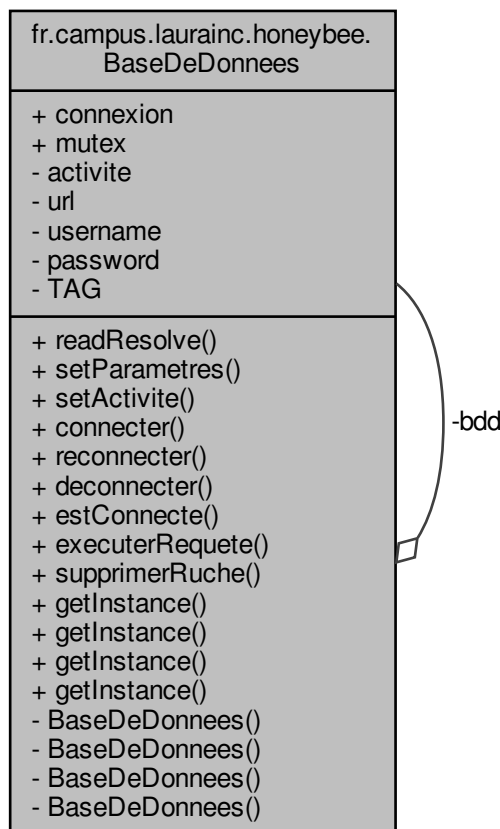
La documentation de cette classe a été générée à partir des fichiers suivants :

- [baseDeDonnees.h](#)
- [baseDeDonnees.cpp](#)

## 9.4 Référence de la classe fr.campus.laurainc.honeybee.BaseDeDonnees

Gestion d'une base de données MySQL (Singleton)

Graphes de collaboration de fr.campus.laurainc.honeybee.BaseDeDonnees :



## Classes

- class [ConnexionMySql](#)  
*Classe permettant de se connecter à MySQL en arrière-plan.*

## Fonctions membres publiques

- Object [readResolve](#) ()  
*Retourne l'objet dans un flux (cf. Singleton)*
- void [setParametres](#) (String [url](#), String [username](#), String [password](#))  
*Modifie les paramètres de connexion à la base de données et se reconnecte.*
- void [setActivite](#) (Activity [activite](#))  
*Fixe l'activité appelante associée à l'instance de Base de données.*
- boolean [connecter](#) ()  
*Connexion à la base de données MySQL si pas déjà connecté*
- boolean [reconnecter](#) ()  
*Reconnexion à la base de données MySQL.*
- boolean [deconnecter](#) ()  
*Déconnexion de la base de données MySQL.*
- boolean [estConnecte](#) ()  
*Retourne vrai si on est connecté à la base de données MySQL.*
- void [executerRequete](#) (final String requete)  
*Méthode qui exécute une requête UPDATE, INSERT ou DELETE en arrière-plan.*
- void [supprimerRuche](#) (final int idRuche)

### Fonctions membres publiques statiques

- static synchronized [BaseDeDonnees getInstance](#) ()  
*Retourne l'instance [BaseDeDonnees](#).*
- static synchronized [BaseDeDonnees getInstance](#) (Activity [activeite](#))
- static synchronized [BaseDeDonnees getInstance](#) (String [url](#), String [username](#), String [password](#))
- static synchronized [BaseDeDonnees getInstance](#) (Activity [activeite](#), String [url](#), String [username](#), String [password](#))

### Attributs publics statiques

- static Connection [connexion](#) = null  
*objet de connexion à MySQL (null par défaut)*
- static final Lock [mutex](#) = new ReentrantLock(true)  
*mutex pour l'exécution concurrente de requêtes*

### Fonctions membres privées

- [BaseDeDonnees](#) ()  
*Constructeur par défaut de la classe [BaseDeDonnees](#).*
- [BaseDeDonnees](#) (Activity [activeite](#))
- [BaseDeDonnees](#) (String [url](#), String [username](#), String [password](#))
- [BaseDeDonnees](#) (Activity [activeite](#), String [url](#), String [username](#), String [password](#))

### Attributs privés

- Activity [activeite](#) = null  
*objet sur l'activeite ayant créé l'objet [BaseDeDonnees](#) (si besoin pour UI)*
- String [url](#)  
*l'URL pointant sur la base de données d'un serveur MySQL*
- String [username](#)  
*le nom du compte utilisateur (root par défaut)*
- String [password](#)  
*le mot de passe du compte utilisateur (password par défaut)*

### Attributs privés statiques

- static final String [TAG](#) = "BaseDeDonnees"  
*le TAG de la classe pour les logs*
- static [BaseDeDonnees bdd](#) = null  
*l'instance unique de [BaseDeDonnees](#) (Singleton)*

## 9.4.1 Description détaillée

### Auteur

Thierry Vaira

## 9.4.2 Documentation des constructeurs et destructeur

### 9.4.2.1 [BaseDeDonnees\(\)](#) [ 1 / 4 ]

`BaseDeDonnees.BaseDeDonnees ( ) [private]`

Constructeur de la classe [BaseDeDonnees](#).

## Paramètres

<i>activite</i>	Activity l'activité appelante
<i>url</i>	String l'URL pointant sur la base de données d'un serveur MySQL
<i>username</i>	String le nom d'utilisateur
<i>password</i>	String le mot de passe de l'utilisateur
<i>activite</i>	Activity l'activité appelante
<i>url</i>	String l'URL pointant sur la base de données d'un serveur MySQL
<i>username</i>	String le nom d'utilisateur
<i>password</i>	String le mot de passe de l'utilisateur

Référencé par [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#).

```
00107    {
00108    }
```

## 9.4.2.2 BaseDeDonnees() [2/4]

```
fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees (
    Activity activite ) [private]
```

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.activite](#).

```
00116    {
00117        this.activite = activite;
00118    }
```

## 9.4.2.3 BaseDeDonnees() [3/4]

```
fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees (
    String url,
    String username,
    String password ) [private]
```

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.password](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.url](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.username](#).

```
00128    {
00129        Log.v(TAG, "url=" + url);
00130        Log.v(TAG, "username=" + username);
00131        Log.v(TAG, "password=" + password);
00132        this.url = url;
00133        this.username = username;
00134        this.password = password;
00135        // on peut se connecter
00136        connecter();
00137    }
```

#### 9.4.2.4 BaseDeDonnees() [4/4]

```
fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees (
    Activity activite,
    String url,
    String username,
    String password ) [private]
```

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.activite](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.password](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.url](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.username](#).

```
00148      {
00149          Log.v(TAG, "url=" + url);
00150          Log.v(TAG, "username=" + username);
00151          Log.v(TAG, "password=" + password);
00152          this.activite = activite;
00153          this.url = url;
00154          this.username = username;
00155          this.password = password;
00156          // on peut se connecter
00157          connecter();
00158      }
```

### 9.4.3 Documentation des fonctions membres

#### 9.4.3.1 connecter()

```
BaseDeDonnees.connecter ( )
```

##### Renvoie

boolean vrai si la connexion a réussi sinon faux

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees\(\)](#), [fr.campus.laurainc.honeybee.MainActivity.onCreate\(\)](#), [fr.campus.laurainc.honeybee.NouvelleRucheActivity.onCreate\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.reconnecter\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.Ruche\(\)](#).

```
00192      {
00193          if(!HoneyBee.BDD)
00194              return false;
00195
00196          try
00197          {
00198              if (!estConnecte())
00199              {
00200                  ConnexionMySQL connexionMySQL = new ConnexionMySQL();
00201                  connexionMySQL.execute();
00202                  Log.d(TAG, "connecter -> retour : " + connexionMySQL.get());
00203                  return connexionMySQL.get();
00204              }
00205              return true;
00206          }
00207          catch (InterruptedException e)
00208          {
00209              e.printStackTrace();
00210              Log.e(TAG, "connecter -> InterruptedException : " + e.toString());
00211              return false;
00212          }
00213          catch (ExecutionException e)
00214          {
00215              e.printStackTrace();
00216              Log.e(TAG, "connecter -> ExecutionException : " + e.toString());
00217              return false;
00218          }
00219      }
```



## 9.4.3.2 deconnecter()

```
BaseDeDonnees.deconnecter ( )
```

## Renvoie

boolean vrai si la déconnexion a réussi sinon faux

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#).

Référencé par [fr.campus.laurainc.honeybee.BaseDeDonnees.reconnecter\(\)](#).

```
00240     {
00241         if (!HoneyBee.BDD)
00242             return false;
00243
00244         try
00245         {
00246             if (connexion != null)
00247             {
00248                 connexion.close();
00249                 Log.d(TAG, "deconnecter -> retour : " + connexion.isClosed());
00250                 return connexion.isClosed();
00251             }
00252             return true;
00253         }
00254         catch (SQLException e)
00255         {
00256             e.printStackTrace();
00257             Log.e(TAG, "deconnecter -> ExecutionException : " + e.toString());
00258             return false;
00259         }
00260     }
```

## 9.4.3.3 estConnecte()

```
BaseDeDonnees.estConnecte ( )
```

## Renvoie

boolean vrai si on est connecté à la base de données MySQL sinon faux

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#).

Référencé par [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.<ConnexionMySQL.doInBackground\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.executerRequete\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recuperer\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererChoixChAppID\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererHistoriqueAlertes\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererIdTTN\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererListeRuches\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererMesuresJournalieresRuche\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererMoyennes\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.supprimerRuche\(\)](#).

```
00268     {
00269         if (!HoneyBee.BDD)
00270             return false;
00271
00272         try
00273         {
00274             if (connexion != null)
00275                 return !connexion.isClosed();
00276             else
00277                 return false;
00278         }
00279         catch (SQLException e)
00280         {
00281             e.printStackTrace();
00282             return false;
00283         }
00284     }
```

#### 9.4.3.4 executerRequete()

```
BaseDeDonnees.executerRequete (
    final String requete )
```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.onCreate\(\)](#).

```
00390      {
00391          if(!HoneyBee.BDD)
00392              return;
00393
00394          if(estConnecte())
00395          {
00396              // Seulement pour les requêtes UPDATE, INSERT ou DELETE
00397              if(!requete.contains("UPDATE") && !requete.contains("INSERT") && !requete.contains("DELETE"))
00398                  return;
00399
00400              // Exécute la requête dans un thread
00401              Thread requeteBDD = new Thread(new Runnable()
00402              {
00403                  public void run()
00404                  {
00405                      mutex.lock();
00406                      try
00407                      {
00408                          Log.d(TAG, "executerRequete -> requete : " + requete);
00409                          Statement statement = connexion.createStatement();
00410                          statement.executeUpdate(requete);
00411                      }
00412                      catch (Exception e)
00413                      {
00414                          e.printStackTrace();
00415                          Log.e(TAG, "executerRequete -> exception : " + e.toString());
00416                      }
00417                      finally
00418                      {
00419                          mutex.unlock();
00420                      }
00421                  }
00422              });
00423
00424              // Démarrage de l'exécution de la requête
00425              Log.v("BDD", "Requete OK = " + requete);
00426              requeteBDD.start();
00427          }
00428          else
00429          {
00430              Log.w(TAG, "Pas de connexion MySQL !");
00431          }
00432      }
```

#### 9.4.3.5 getInstance() [1/4]

```
BaseDeDonnees.getInstance ( ) [static]
```

##### Renvoie

[BaseDeDonnees](#) l'instance [BaseDeDonnees](#)

##### Paramètres

<i>activité</i>	Activity l'activité appelante
-----------------	-------------------------------

##### Renvoie

[BaseDeDonnees](#) l'instance [BaseDeDonnees](#)

## Paramètres

<i>url</i>	String l'URL pointant sur la base de données d'un serveur MySQL
<i>username</i>	String le nom d'utilisateur
<i>password</i>	String le mot de passe de l'utilisateur

## Renvoie

[BaseDeDonnees](#) l'instance [BaseDeDonnees](#)

## Paramètres

<i>activite</i>	Activity l'activité appelante
<i>url</i>	String l'URL pointant sur la base de données d'un serveur MySQL
<i>username</i>	String le nom d'utilisateur
<i>password</i>	String le mot de passe de l'utilisateur

## Renvoie

[BaseDeDonnees](#) l'instance [BaseDeDonnees](#)

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.bdd](#).

Référencé par [fr.campus.laurainc.honeybee.MainActivity.onCreate\(\)](#), [fr.campus.laurainc.honeybee.NouvelleRucheActivity.onCreate\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.readResolve\(\)](#), [fr.campus.laurainc.honeybee.Ruche.Ruche\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.supprimerRuche\(\)](#).

```
00042    {
00043        if(bdd == null)
00044            bdd = new BaseDeDonnees();
00045        return bdd;
00046    }
```

## 9.4.3.6 getInstance() [2/4]

```
static synchronized BaseDeDonnees fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance (
    Activity activite ) [static]
```

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.bdd](#).

```
00055    {
00056        if(bdd == null)
00057            bdd = new BaseDeDonnees(activite);
00058        return bdd;
00059    }
```

### 9.4.3.7 getInstance() [3/4]

```
static synchronized BaseDeDonnees fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance (
    String url,
    String username,
    String password ) [static]
```

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.bdd](#).

```
00070      {
00071          if(bdd == null)
00072              bdd = new BaseDeDonnees(url, username,
00073          password);
00073          return bdd;
00074      }
```

### 9.4.3.8 getInstance() [4/4]

```
static synchronized BaseDeDonnees fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance (
    Activity activite,
    String url,
    String username,
    String password ) [static]
```

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.bdd](#).

```
00086      {
00087          if(bdd == null)
00088              bdd = new BaseDeDonnees(activite, url,
00089          username, password);
00089          return bdd;
00090      }
```

### 9.4.3.9 readResolve()

```
BaseDeDonnees.readResolve ( )
```

#### Renvoie

[BaseDeDonnees](#) l'instance [BaseDeDonnees](#)

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#).

```
00098      {
00099          return BaseDeDonnees.getInstance();
00100      }
```

## 9.4.3.10 reconnecter()

```
BaseDeDonnees.reconnecter ( )
```

## Renvoie

boolean vrai si la connexion a réussi sinon faux

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.deconnecter\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.MainActivity.onActivityResult\(\)](#).

```
00227    {
00228        if (!HoneyBee.BDD)
00229            return false;
00230        deconnecter();
00231        return connecter();
00232    }
```

## 9.4.3.11 setActivite()

```
BaseDeDonnees.setActivite (
    Activity activite )
```

## Paramètres

<i>activite</i>	Activity l'activité appelante
-----------------	-------------------------------

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.activite](#).

```
00182    {
00183        this.activite = activite;
00184    }
```

## 9.4.3.12 setParametres()

```
BaseDeDonnees.setParametres (
    String url,
    String username,
    String password )
```

## Paramètres

<i>url</i>	String l'URL pointant sur la base de données d'un serveur MySQL
<i>username</i>	String le nom d'utilisateur
<i>password</i>	String le mot de passe de l'utilisateur

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.password](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.url](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.username](#).

Référencé par [fr.campus.laurainc.honeybee.MainActivity.onActivityResult\(\)](#).

```
00168    {
00169        this.url = url;
00170        this.username = username;
00171        this.password = password;
00172        // reconnexion
00173        //reconnecter();
00174    }
```

#### 9.4.3.13 supprimerRuche()

```
void fr.campus.laurainc.honeybee.BaseDeDonnees.supprimerRuche (
    final int idRuche )
```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.supprimerRuche\(\)](#).

```
00434    {
00435        if (!HoneyBee.BDD)
00436            return;
00437
00438        final String requete = "DELETE FROM Ruche WHERE idRuche='" + idRuche + "'";
00439        if (estConnecte()) {
00440            Thread requeteBDD = new Thread(new Runnable() {
00441                public void run() {
00442
00443                    mutex.lock();
00444                    try {
00445                        Log.d(TAG, "executerRequete -> requete : " + requete);
00446                        Statement statement = connexion.createStatement();
00447                        statement.executeUpdate(requete);
00448                    } catch (Exception e) {
00449                        e.printStackTrace();
00450                        Log.e(TAG, "executerRequete -> exception : " + e.toString());
00451                    } finally {
00452                        mutex.unlock();
00453                    }
00454                }
00455            });
00456
00457            Log.v("BDD", "Requete OK = " + requete);
00458            requeteBDD.start();
00459        } else {
00460            Log.w(TAG, "Pas de connexion MySQL !");
00461        }
00462    }}
```

### 9.4.4 Documentation des données membres

#### 9.4.4.1 activite

```
Activity fr.campus.laurainc.honeybee.BaseDeDonnees.activite = null [private]
```

Référencé par [fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.setActivite\(\)](#).

#### 9.4.4.2 bdd

`BaseDeDonnees` fr.campus.laurainc.honeybee.BaseDeDonnees.bdd = null [static], [private]

Référencé par `fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance()`.

#### 9.4.4.3 connexion

`Connection` fr.campus.laurainc.honeybee.BaseDeDonnees.connexion = null [static]

Référencé par `fr.campus.laurainc.honeybee.Ruche.recuperer()`, `fr.campus.laurainc.honeybee.Ruche.recupererChoixChAppID()`, `fr.campus.laurainc.honeybee.Ruche.recupererHistoriqueAlertes()`, `fr.campus.laurainc.honeybee.Ruche.recupererIdTTN()`, `fr.campus.laurainc.honeybee.Ruche.recupererListeRuches()`, `fr.campus.laurainc.honeybee.Ruche.recupererMesuresJournalieresRuche()`, et `fr.campus.laurainc.honeybee.Ruche.recupererMoyennes()`.

#### 9.4.4.4 mutex

`final Lock` fr.campus.laurainc.honeybee.BaseDeDonnees.mutex = new ReentrantLock(true) [static]

Référencé par `fr.campus.laurainc.honeybee.Ruche.recuperer()`, `fr.campus.laurainc.honeybee.Ruche.recupererChoixChAppID()`, `fr.campus.laurainc.honeybee.Ruche.recupererHistoriqueAlertes()`, `fr.campus.laurainc.honeybee.Ruche.recupererIdTTN()`, `fr.campus.laurainc.honeybee.Ruche.recupererListeRuches()`, `fr.campus.laurainc.honeybee.Ruche.recupererMesuresJournalieresRuche()`, et `fr.campus.laurainc.honeybee.Ruche.recupererMoyennes()`.

#### 9.4.4.5 password

`String` fr.campus.laurainc.honeybee.BaseDeDonnees.password [private]

Référencé par `fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees()`, et `fr.campus.laurainc.honeybee.BaseDeDonnees.setParametres()`.

#### 9.4.4.6 TAG

`final String` fr.campus.laurainc.honeybee.BaseDeDonnees.TAG = "BaseDeDonnees" [static], [private]

#### 9.4.4.7 url

`String` fr.campus.laurainc.honeybee.BaseDeDonnees.url [private]

Référencé par `fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees()`, et `fr.campus.laurainc.honeybee.BaseDeDonnees.setParametres()`.

#### 9.4.4.8 username

```
String fr.campus.laurainc.honeybee.BaseDeDonnees.username [private]
```

Référencé par [fr.campus.laurainc.honeybee.BaseDeDonnees.BaseDeDonnees\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.setParametres\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

— [BaseDeDonnees.java](#)

### 9.5 Référence de la classe fr.campus.laurainc.honeybee.CarteActivity

Graphe de collaboration de fr.campus.laurainc.honeybee.CarteActivity :

fr.campus.laurainc.honeybee. CarteActivity
~ TAG - mMap
+ onMapReady() # onCreate()

#### Fonctions membres publiques

— void [onMapReady](#) (GoogleMap googleMap)

#### Fonctions membres protégées

— void [onCreate](#) (Bundle savedInstanceState)

#### Attributs privés

— GoogleMap [mMap](#)

#### 9.5.1 Documentation des fonctions membres



## 9.5.1.1 onCreate()

```

void fr.campus.laurainc.honeybee.CarteActivity.onCreate (
    Bundle savedInstanceState ) [protected]

00033                                     {
00034     super.onCreate(savedInstanceState);
00035     setContentView(R.layout.activity_carte);
00036
00037
00038     // Obtain the SupportMapFragment and get notified when the map is ready to be used.
00039     SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
00040         .findFragmentById(R.id.map);
00041     mapFragment.getMapAsync(this);
00042 }

```

## 9.5.1.2 onMapReady()

```

void fr.campus.laurainc.honeybee.CarteActivity.onMapReady (
    GoogleMap googleMap )

```

Manipulates the map once available. This callback is triggered when the map is ready to be used. This is where we can add markers or lines, add listeners or move the camera. In this case, we just add a marker near Sydney, Australia. If Google Play services is not installed on the device, the user will be prompted to install it inside the SupportMapFragment. This method will only be triggered once the user has installed Google Play services and returned to the app.

```

00055                                     {
00056     mMap = googleMap;
00057
00058     Intent intent = getIntent();
00059
00060     double latitude = 0.0;
00061     if (intent.hasExtra("latitude")){ // vérifie qu'une valeur est associée à la clé "edittext"
00062         latitude = Double.parseDouble(intent.getStringExtra("latitude").replaceAll("°", "")); // on
récupère la valeur associée à la clé
00063         Log.v("Map", "Latitude :" + valueOf(latitude));
00064     }
00065
00066     double longitude = 0.0;
00067     if (intent.hasExtra("longitude")){ // vérifie qu'une valeur est associée à la clé "edittext"
00068         longitude = Double.parseDouble(intent.getStringExtra("longitude").replaceAll("°", "")); // on
récupère la valeur associée à la clé
00069         Log.v("Map", "Longitude" + valueOf(longitude));
00070     }
00071
00072     // Add a marker in Sydney and move the camera
00073     LatLng sydney = new LatLng(latitude, longitude);
00074     mMap.addMarker(new MarkerOptions().position(sydney).title("Ruche"));
00075     mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
00076     mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, 15));
00077     mMap.animateCamera(CameraUpdateFactory.zoomTo(10), 2000, null);
00078     Circle circle = mMap.addCircle(new CircleOptions()
00079         .center(new LatLng(latitude, longitude))
00080         .radius(10000)
00081         .strokeColor(Color.RED));
00082 }

```

## 9.5.2 Documentation des données membres

## 9.5.2.1 mMap

```

GoogleMap fr.campus.laurainc.honeybee.CarteActivity.mMap [private]

```

La documentation de cette classe a été générée à partir du fichier suivant :

— [CarteActivity.java](#)

## 9.6 Référence de la classe fr.campus.laurainc.honeybee.ClientMQTT

Grphe de collaboration de fr.campus.laurainc.honeybee.ClientMQTT :

fr.campus.laurainc.honeybee. ClientMQTT
+ mqttAndroidClient + PORT_MESURE_ENERGIE + PORT_MESURE_POIDS + PORT_MESURE_RUCHE + PORT_MESURE_ENVIRONEMENT + PORT_MESURE_ENSOLEILLEMENT - serverUri - clientId - username - password - subscriptionTopic - connecte - inscrit - TAG
+ ClientMQTT() + setCallback() + deconnecter() + souscrireTopic() + desinscrireTopic() + estConnecte() + estInscrit() + setSubscriptionTopic() + extraireHumiditeInterieure() + extraireTemperatureInterieure() et 8 de plus... - connecter()

### Fonctions membres publiques

- [ClientMQTT](#) (String deviceId, String topic, Context context)
- void [setCallback](#) (MqttCallbackExtended callback)
- void [deconnecter](#) ()
- void [souscrireTopic](#) ()
- void [desinscrireTopic](#) ()
- boolean [estConnecte](#) ()
- boolean [estInscrit](#) ()
- void [setSubscriptionTopic](#) (String [subscriptionTopic](#))
- double [extraireHumiditeInterieure](#) (MqttMessage mqttMessage)
- double [extraireTemperatureInterieure](#) (MqttMessage mqttMessage)
- int [extrairePoids](#) (MqttMessage mqttMessage)
- String [extraireHorodatage](#) (MqttMessage mqttMessage)
- double [extraireTemperatureExterieur](#) (MqttMessage mqttMessage)
- double [extraireHumiditeExterieur](#) (MqttMessage mqttMessage)
- double [extrairePression](#) (MqttMessage mqttMessage)
- double [extraireEnsoleillement](#) (MqttMessage mqttMessage)
- int [extraireCharge](#) (MqttMessage mqttMessage)
- double [arrondir](#) (double nombre, double nbApresVirgule)

### Attributs publics

- MqttAndroidClient [mqttAndroidClient](#)

## Attributs publics statiques

- static final int `PORT_MESURE_ENERGIE` = 1
- static final int `PORT_MESURE_POIDS` = 2
- static final int `PORT_MESURE_RUCHE` = 3
- static final int `PORT_MESURE_ENVIRONNEMENT` = 4
- static final int `PORT_MESURE_ENSOLEILLEMENT` = 5

## Fonctions membres privées

- void `connecter` ()

## Attributs privés

- String `serverUri` = "tcp ://eu.thethings.network :1883"
- String `clientId` = "mes\_ruches"
- String `username` = "mes\_ruches"
- String `password` = "ttn-account-v2.vC-aqMRnLLzGkNjODWgy81kLqzxBPAT8\_mE-L7U2C\_w"
- String `subscriptionTopic`
- boolean `connecte`
- boolean `inscrit`

## Attributs privés statiques

- static final String `TAG` = "ClientMQTT"  
le TAG de la classe pour les logs

## 9.6.1 Documentation des constructeurs et destructeur

## 9.6.1.1 ClientMQTT()

```
fr.campus.laurainc.honeybee.ClientMQTT.ClientMQTT (
    String deviceID,
    String topic,
    Context context )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.connecter\(\)](#), et [fr.campus.laurainc.honeybee.ClientMQTT.setSubscription←Topic\(\)](#).

```
00039    {
00040        Log.v(TAG, "deviceID : " + deviceID + " topic : " + topic);
00041        setSubscriptionTopic(topic);
00042        connecte = false;
00043        inscrit = false;
00044        mqttAndroidClient = new MqttAndroidClient(context,
serverUri, clientId);
00045        mqttAndroidClient.setCallback(new MqttCallbackExtended()
00046        {
00047            @Override
00048            public void connectComplete(boolean b, String s) {
00049                Log.w(TAG, s);
00050            }
00051
00052            @Override
00053            public void connectionLost(Throwable throwable) {
00054                Log.w(TAG, "Connexion perdue avec le serveur");
00055                connecte = false;
00056            }
00057
00058            @Override
00059            public void messageArrived(String topic, MqttMessage mqttMessage) throws Exception {
00060                Log.w(TAG, mqttMessage.toString());
00061            }
00062
00063            @Override
00064            public void deliveryComplete(IMqttDeliveryToken iMqttDeliveryToken) {Log.w("mqtt", "Message
envoyé");}
00065        });
00066
00067        connecter();
00068    }
```

## 9.6.2 Documentation des fonctions membres

### 9.6.2.1 arrondir()

```
double fr.campus.laurainc.honeybee.ClientMQTT.arrondir (
    double nombre,
    double nbAprèsVirgule )
```

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherPoids\(\)](#).

```
00408      {
00409          return(double)((int)(nombre * Math.pow(10,nbAprèsVirgule) + .5)) / Math.pow(10,nbAprèsVirgule);
00410      }
```

### 9.6.2.2 connecter()

```
void fr.campus.laurainc.honeybee.ClientMQTT.connecter ( ) [private]
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.souscrireTopic\(\)](#).

Référéncé par [fr.campus.laurainc.honeybee.ClientMQTT.ClientMQTT\(\)](#).

```
00074      {
00075          MqttConnectOptions mqttConnectOptions = new MqttConnectOptions();
00076          mqttConnectOptions.setAutomaticReconnect(true);
00077          mqttConnectOptions.setCleanSession(false);
00078          mqttConnectOptions.setUserName(username);
00079          mqttConnectOptions.setPassword(password.toCharArray());
00080
00081          try {
00082              Log.d(TAG, "Connexion au serveur : " + serverUri);
00083              mqttAndroidClient.connect(mqttConnectOptions, null, new IMqttActionListener()
00084          {
00085              @Override
00086              public void onSuccess(IMqttToken asyncActionToken) {
00087                  DisconnectedBufferOptions disconnectedBufferOptions = new DisconnectedBufferOptions();
00088                  disconnectedBufferOptions.setBufferEnabled(true);
00089                  disconnectedBufferOptions.setBufferSize(100);
00090                  disconnectedBufferOptions.setPersistBuffer(false);
00091                  disconnectedBufferOptions.setDeleteOldestMessages(false);
00092                  mqttAndroidClient.setBufferOpts(disconnectedBufferOptions);
00093                  connecte = true;
00094                  souscrireTopic();
00095                  Log.d(TAG, "Connecté au serveur : " + serverUri);
00096              }
00097              @Override
00098              public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
00099                  Log.d(TAG, "Impossible de se connecter au serveur : " +
00100serverUri + exception.toString());
00101              }
00102          });
00103          } catch (MqttException ex){
00104              ex.printStackTrace();
00105          } // Méthode permettant de se connecter
```

## 9.6.2.3 deconnecter()

```
void fr.campus.laurainc.honeybee.ClientMQTT.deconnecter ( )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.desinscrireTopic\(\)](#), et [fr.campus.laurainc.honeybee.ClientMQTT.estInscrit\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherListeRuches\(\)](#).

```
00107         {
00108             try {
00109                 if(estInscrit())
00110                     desinscrireTopic();
00111
00112                 IMqttToken disconToken = mqttAndroidClient.disconnect();
00113                 disconToken.setActionCallback(new IMqttActionListener() {
00114                     @Override
00115                     public void onSuccess(IMqttToken asyncActionToken) {
00116                         Log.d(TAG, "Déconnecté du serveur : " + serverUri);
00117                         connecte = false;
00118                     }
00119
00120                     @Override
00121                     public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
00122                         Log.d(TAG, "Impossible de se déconnecter du serveur : " +
00123                             serverUri + exception.toString());
00124                     }
00125                 });
00126             } catch (MqttException e) {
00127                 e.printStackTrace();
00128             }
00129         }
```

## 9.6.2.4 desinscrireTopic()

```
void fr.campus.laurainc.honeybee.ClientMQTT.desinscrireTopic ( )
```

Référencé par [fr.campus.laurainc.honeybee.ClientMQTT.deconnecter\(\)](#).

```
00152         {
00153             try {
00154                 IMqttToken unsubToken = mqttAndroidClient.unsubscribe(
00155                     subscriptionTopic);
00156                 unsubToken.setActionCallback(new IMqttActionListener() {
00157                     @Override
00158                     public void onSuccess(IMqttToken asyncActionToken) {
00159                         Log.w(TAG, "Désabonné du topic " + subscriptionTopic);
00160                         inscrit = false;
00161                     }
00162
00163                     @Override
00164                     public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
00165                         Log.w(TAG, "Impossible de se désabonner du topic");
00166                     }
00167                 });
00168             } catch (MqttException e) {
00169                 e.printStackTrace();
00170             }
00171         }
```

## 9.6.2.5 estConnecte()

```
boolean fr.campus.laurainc.honeybee.ClientMQTT.estConnecte ( )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.connecte](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherListeRuches\(\)](#).

```
00172         {
00173             Log.w(TAG, "isConnected " + mqttAndroidClient.isConnected());
00174             Log.w(TAG, "connecte " + connecte);
00175             //return mqttAndroidClient.isConnected();
00176             return connecte;
00177         }
```

### 9.6.2.6 estInscrit()

```
boolean fr.campus.laurainc.honeybee.ClientMQTT.estInscrit ( )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.inscrit](#).

Référencé par [fr.campus.laurainc.honeybee.ClientMQTT.deconnecter\(\)](#).

```
00179         {
00180             Log.w(TAG,"inscrit " + inscrit);
00181             return inscrit;
00182         }
```

### 9.6.2.7 extraireCharge()

```
int fr.campus.laurainc.honeybee.ClientMQTT.extraireCharge (
    MqttMessage mqttMessage )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00364     {
00365         final String message = mqttMessage.toString();
00366         int charge = 0;
00367         try
00368         {
00369             JSONObject jsonObjet = new JSONObject(message);
00370             JSONObject payloadFields = jsonObjet.getJSONObject("payload_fields");
00371             charge = payloadFields.getInt("charge");
00372             Log.d(TAG, "Charge : " + charge + " %");
00373             Log.d(TAG, "Charge : " + charge + " %");
00374             Log.d(TAG, "Charge : " + charge + " %");
00375             extraireHorodatage(mqttMessage);
00376         }
00377         catch (JSONException e)
00378         {
00379             e.printStackTrace();
00380         }
00381         return charge;
00382     }
00383 }
```

### 9.6.2.8 extraireEnsoleillement()

```
double fr.campus.laurainc.honeybee.ClientMQTT.extraireEnsoleillement (
    MqttMessage mqttMessage )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00342     {
00343         final String message = mqttMessage.toString();
00344         double ensoleillement = 0.;
00345         try
00346         {
00347             JSONObject jsonObjet = new JSONObject(message);
00348             JSONObject payloadFields = jsonObjet.getJSONObject("payload_fields");
00349             ensoleillement = payloadFields.getDouble("ensoleillement");
00350             Log.d(TAG, "Ensoleillement : " + ensoleillement + " hPa");
00351             Log.d(TAG, "Ensoleillement : " + ensoleillement + " hPa");
00352             extraireHorodatage(mqttMessage);
00353         }
00354         catch (JSONException e)
00355         {
00356             e.printStackTrace();
00357         }
00358         return ensoleillement;
00359     }
00360 }
00361 }
```

## 9.6.2.9 extraireHorodatage()

```
String fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage (
    MqttMessage mqttMessage )
```

Référencé par [fr.campus.laurainc.honeybee.ClientMQTT.extraireCharge\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireEnsoleillement\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireHumiditeExterieur\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireHumiditeInterieur\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extrairePoids\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extrairePression\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireTemperatureExterieur\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireTemperatureInterieur\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00255     {
00256         final String message = mqttMessage.toString();
00257         String date = "";
00258
00259         try
00260         {
00261             JSONObject jsonObject = new JSONObject(message);
00262             date = jsonObject.getJSONObject("metadata").getString("time");
00263             date = date.substring(0, 10) + " " + date.substring(11, 19);
00264             Log.d(TAG, "Horodatage : " + date);
00265         }
00266         catch (JSONException e)
00267         {
00268             e.printStackTrace();
00269         }
00270
00271         return date;
00272     }
00273 }
```

## 9.6.2.10 extraireHumiditeExterieur()

```
double fr.campus.laurainc.honeybee.ClientMQTT.extraireHumiditeExterieur (
    MqttMessage mqttMessage )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00298     {
00299         final String message = mqttMessage.toString();
00300         double humiditeExterieur = 0.;
00301
00302         try
00303         {
00304             JSONObject jsonObject = new JSONObject(message);
00305             JSONObject payloadFields = jsonObject.getJSONObject("payload_fields");
00306             humiditeExterieur = payloadFields.getDouble("humidite");
00307             Log.d(TAG, "Humidité extérieure : " + humiditeExterieur + " %");
00308
00309             extraireHorodatage(mqttMessage);
00310         }
00311         catch (JSONException e)
00312         {
00313             e.printStackTrace();
00314         }
00315
00316         return humiditeExterieur;
00317     }
```

### 9.6.2.11 extraireHumiditeInterieure()

```
double fr.campus.laurainc.honeybee.ClientMQTT.extraireHumiditeInterieure (
    MqttMessage mqttMessage )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#).

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00189     {
00190         final String message = mqttMessage.toString();
00191         double humidite = 0.;
00192
00193         try
00194         {
00195             JSONObject jsonObjet = new JSONObject(message);
00196             JSONObject payloadFields = jsonObjet.getJSONObject("payload_fields");
00197             humidite = payloadFields.getDouble("humidite");
00198             Log.d(TAG, "Humidite intérieure : " + humidite + " %");
00199
00200             extraireHorodatage(mqttMessage);
00201         }
00202         catch (JSONException e)
00203         {
00204             e.printStackTrace();
00205         }
00206
00207         return humidite;
00208     }
```

### 9.6.2.12 extrairePoids()

```
int fr.campus.laurainc.honeybee.ClientMQTT.extrairePoids (
    MqttMessage mqttMessage )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#).

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00233     {
00234         final String message = mqttMessage.toString();
00235         int poids = 0;
00236
00237         try
00238         {
00239             JSONObject jsonObjet = new JSONObject(message);
00240             JSONObject payloadFields = jsonObjet.getJSONObject("payload_fields");
00241             poids = payloadFields.getInt("poids");
00242             Log.d(TAG, "Poids : " + poids + " Kg");
00243
00244             extraireHorodatage(mqttMessage);
00245         }
00246         catch (JSONException e)
00247         {
00248             e.printStackTrace();
00249         }
00250
00251         return poids;
00252     }
```



### 9.6.2.13 extrairePression()

```
double fr.campus.laurainc.honeybee.ClientMQTT.extrairePression (
    MqttMessage mqttMessage )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#).

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00320     {
00321         final String message = mqttMessage.toString();
00322         double pression = 0.;
00323
00324         try
00325         {
00326             JSONObject jsonObjet = new JSONObject(message);
00327             JSONObject payloadFields = jsonObjet.getJSONObject("payload_fields");
00328             pression = payloadFields.getDouble("pression");
00329             Log.d(TAG, "Pression : " + pression + " hPa");
00330
00331             extraireHorodatage(mqttMessage);
00332         }
00333         catch (JSONException e)
00334         {
00335             e.printStackTrace();
00336         }
00337
00338         return pression;
00339     }
```

### 9.6.2.14 extraireTemperatureExterieur()

```
double fr.campus.laurainc.honeybee.ClientMQTT.extraireTemperatureExterieur (
    MqttMessage mqttMessage )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#).

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00276     {
00277         final String message = mqttMessage.toString();
00278         double temperatureExterieur = 0.;
00279
00280         try
00281         {
00282             JSONObject jsonObjet = new JSONObject(message);
00283             JSONObject payloadFields = jsonObjet.getJSONObject("payload_fields");
00284             temperatureExterieur = payloadFields.getDouble("temperature");
00285             Log.d(TAG, "Temperature extérieure : " + temperatureExterieur + " °C");
00286
00287             extraireHorodatage(mqttMessage);
00288         }
00289         catch (JSONException e)
00290         {
00291             e.printStackTrace();
00292         }
00293
00294         return temperatureExterieur;
00295     }
```

### 9.6.2.15 extraireTemperatureInterieure()

```
double fr.campus.laurainc.honeybee.ClientMQTT.extraireTemperatureInterieure (
    MqttMessage mqttMessage )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#).

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00211     {
00212         final String message = mqttMessage.toString();
00213         double temperature = 0.;
00214
00215         try
00216         {
00217             JSONObject jsonObjet = new JSONObject(message);
00218             JSONObject payloadFields = jsonObjet.getJSONObject("payload_fields");
00219             temperature = payloadFields.getDouble("temperature");
00220             Log.d(TAG, "Température : " + temperature + " °C");
00221
00222             extraireHorodatage(mqttMessage);
00223         }
00224         catch (JSONException e)
00225         {
00226             e.printStackTrace();
00227         }
00228
00229         return temperature;
00230     }
```

### 9.6.2.16 setCallback()

```
void fr.campus.laurainc.honeybee.ClientMQTT.setCallback (
    MqttCallbackExtended callback )
```

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivitycommuniquerTTN\(\)](#).

```
00070                                     {
00071         mqttAndroidClient.setCallback(callback);
00072     }
```

### 9.6.2.17 setSubscriptionTopic()

```
void fr.campus.laurainc.honeybee.ClientMQTT.setSubscriptionTopic (
    String subscriptionTopic )
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.subscriptionTopic](#).

Référéncé par [fr.campus.laurainc.honeybee.ClientMQTT.ClientMQTT\(\)](#).

```
00184                                     {
00185         this.subscriptionTopic = subscriptionTopic;
00186     }
```

## 9.6.2.18 souscrireTopic()

```
void fr.campus.laurainc.honeybee.ClientMQTT.souscrireTopic ( )
```

Référencé par [fr.campus.laurainc.honeybee.ClientMQTT.connecter\(\)](#).

```
00130                                     {
00131         try {
00132             mqttAndroidClient.subscribe(subscriptionTopic, 0, null, new
IMqttActionListener() {
00133                 @Override
00134                 public void onSuccess(IMqttToken asyncActionToken) {
00135                     Log.w(TAG, "Abonné au topic " + subscriptionTopic);
00136                     inscrit = true;
00137                 }
00138                 @Override
00139                 public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
00140                     Log.w(TAG, "Impossible de s'abonner du topic");
00141                     inscrit = false;
00142                 }
00143             });
00144         } catch (MqttException ex) {
00145             System.err.println("Exceptionst subscribing");
00146             ex.printStackTrace();
00147         }
00148     } // Abonnement à un topic
```

## 9.6.3 Documentation des données membres

## 9.6.3.1 clientId

```
String fr.campus.laurainc.honeybee.ClientMQTT.clientId = "mes_ruches" [private]
```

## 9.6.3.2 connecte

```
boolean fr.campus.laurainc.honeybee.ClientMQTT.connecte [private]
```

Référencé par [fr.campus.laurainc.honeybee.ClientMQTT.estConnecte\(\)](#).

## 9.6.3.3 inscrit

```
boolean fr.campus.laurainc.honeybee.ClientMQTT.inscrit [private]
```

Référencé par [fr.campus.laurainc.honeybee.ClientMQTT.estInscrit\(\)](#).

## 9.6.3.4 mqttAndroidClient

```
MqttAndroidClient fr.campus.laurainc.honeybee.ClientMQTT.mqttAndroidClient
```

### 9.6.3.5 password

```
String fr.campus.laurainc.honeybee.ClientMQTT.password = "ttn-account-v2.vC-aqMRnLLzGkNjODWgy81kLqzxBPAT8_↔  
mE-L7U2C_w" [private]
```

### 9.6.3.6 PORT\_MESURE\_ENERGIE

```
final int fr.campus.laurainc.honeybee.ClientMQTT.PORT_MESURE_ENERGIE = 1 [static]
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

### 9.6.3.7 PORT\_MESURE\_ENSOLEILLEMENT

```
final int fr.campus.laurainc.honeybee.ClientMQTT.PORT_MESURE_ENSOLEILLEMENT = 5 [static]
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

### 9.6.3.8 PORT\_MESURE\_ENVIRONEMENT

```
final int fr.campus.laurainc.honeybee.ClientMQTT.PORT_MESURE_ENVIRONEMENT = 4 [static]
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

### 9.6.3.9 PORT\_MESURE\_POIDS

```
final int fr.campus.laurainc.honeybee.ClientMQTT.PORT_MESURE_POIDS = 2 [static]
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

### 9.6.3.10 PORT\_MESURE\_RUCHE

```
final int fr.campus.laurainc.honeybee.ClientMQTT.PORT_MESURE_RUCHE = 3 [static]
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

### 9.6.3.11 serverUri

```
String fr.campus.laurainc.honeybee.ClientMQTT.serverUri = "tcp://eu.thethings.network:1883" [private]
```

## 9.6.3.12 subscriptionTopic

```
String fr.campus.laurainc.honeybee.ClientMQTT.subscriptionTopic [private]
```

Référencé par [fr.campus.laurainc.honeybee.ClientMQTT.setSubscriptionTopic\(\)](#).

## 9.6.3.13 TAG

```
final String fr.campus.laurainc.honeybee.ClientMQTT.TAG = "ClientMQTT" [static], [private]
```

## 9.6.3.14 username

```
String fr.campus.laurainc.honeybee.ClientMQTT.username = "mes_ruches" [private]
```

La documentation de cette classe a été générée à partir du fichier suivant :

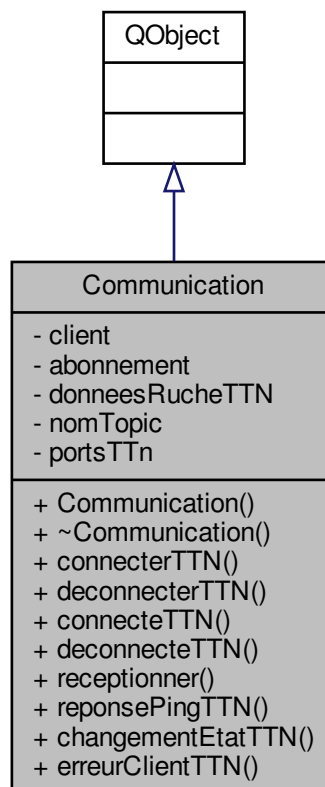
— [ClientMQTT.java](#)

## 9.7 Référence de la classe Communication

La classe [Communication](#).

```
#include <communication.h>
```

Graphe de collaboration de Communication :



## Connecteurs publics

- void `connecteTTN` ()  
*slot permettant la connection au serveur ttn*
- void `deconnecteTTN` ()  
*slot permettant la deconnection au serveur ttn*
- void `receptionner` (const QByteArray &message, const QMqttTopicName &topic)  
*slot permettant la reception des donnée grace au protocole mqtt*
- void `reponsePingTTN` ()  
*slot permettant d'effectuer un ping vers le serveur ttn*
- void `changementEtatTTN` ()  
*slot permettant de recevoir l'etat du serveur ttn*
- void `erreurClientTTN` ()  
*slot permettant de recevoir les erreur rencontré lors de la connection au serveur ttn*

## Signaux

- void `etatClientConnexion` (bool connexion)
- void `messageRecu` (const QByteArray &message, const QMqttTopicName &topic)

## Fonctions membres publiques

- `Communication` (QStringList `donneesRucheTTN`, QObject \*parent=0)  
*Constructeur de la classe `Communication`.*
- `~Communication` ()  
*destructeur de la classe `Communication`*
- void `connecterTTN` ()
- void `deconnecterTTN` ()

## Attributs privés

- QMqttClient \* `client`
- QMqttSubscription \* `abonnement`
- QStringList `donneesRucheTTN`
- QString `nomTopic`
- PortsTTN `portsTTN`

### 9.7.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

### 9.7.2 Documentation des constructeurs et destructeur

#### 9.7.2.1 Communication()

```
Communication::Communication (
    QStringList donneesRucheTTN,
    QObject * parent = 0 )
```

## Paramètres

<i>parent</i>	<a href="#">QObject</a> Adresse de l'objet Qt parent (0 = fenêtre principale)
---------------	---

Définition des attributs client , abonnement et nomTopic a 0 (=null)

Références [changementEtatTTN\(\)](#), [client](#), [connecterTTN\(\)](#), [connecteTTN\(\)](#), [deconnecteTTN\(\)](#), [erreurClientTTN\(\)](#), [nomTopic](#), [receptionner\(\)](#), et [reponsePingTTN\(\)](#).

```

00027                                     :
      QObject(parent), client(0), abonnement(0),
      donneesRucheTTN(donneesRucheTTN)
00028 {
00029     nomTopic = donneesRucheTTN.at(6) + "/devices/" +
      donneesRucheTTN.at(2) + "/up";
00030     qDebug() << Q_FUNC_INFO << "donneesRucheTTN" << donneesRucheTTN << "nomTopic" <<
      nomTopic;
00031     client = new QMqttClient();
00032     connect(client, SIGNAL(stateChanged(ClientState)), this, SLOT(
      changementEtatTTN()));
00033     connect(client, SIGNAL(errorChanged(ClientError)), this, SLOT(
      erreurClientTTN()));
00034     connect(client, SIGNAL(connected()), this, SLOT(connecteTTN()));
00035     connect(client, SIGNAL(disconnected()), this, SLOT(deconnecteTTN()));
00036     connect(client, SIGNAL(messageReceived(const QByteArray &, const QMqttTopicName &)), this, SLOT(
      receptionner(const QByteArray &, const QMqttTopicName &)));
00037     connect(client, SIGNAL(pingResponseReceived()), this, SLOT(
      reponsePingTTN()));
00038     connecterTTN();
00039 }
```

## 9.7.2.2 ~Communication()

Communication::~~Communication ( )

Références [abonnement](#), [client](#), et [deconnecterTTN\(\)](#).

```

00045 {
00046     deconnecterTTN();
00047     delete abonnement;
00048     delete client;
00049     qDebug() << Q_FUNC_INFO;
00050 }
```

## 9.7.3 Documentation des fonctions membres

## 9.7.3.1 changementEtatTTN

void Communication::changementEtatTTN ( ) [slot]

Références [client](#).

Référencé par [Communication\(\)](#).

```

00126 {
00127     // Pour le debug
00128     QString message;
00129     switch(client->state())
00130     {
00131         case 0: message = "Déconnecté"; break;
00132         case 1: message = "En cours de connexion"; break;
00133         case 2: message = "Connecté"; break;
00134     }
00135     qDebug() << Q_FUNC_INFO << "client MQTT" << message << "state=" << client->state();
00136 }
```

### 9.7.3.2 connecterTTN()

```
void Communication::connecterTTN ( )
```

Références [client](#), et [donneesRucheTTN](#).

Référencé par [Communication\(\)](#).

```
00053 {
00054     qDebug() << Q_FUNC_INFO << donneesRucheTTN.at(4) <<
        donneesRucheTTN.at(5).toInt() << donneesRucheTTN.at(6) <<
        donneesRucheTTN.at(7);
00055     client->setHostname(donneesRucheTTN.at(4));
00056     client->setPort(donneesRucheTTN.at(5).toInt());
00057     client->setUsername(donneesRucheTTN.at(6));
00058     client->setPassword(donneesRucheTTN.at(7));
00059     client->connectToHost();
00060 }
```

### 9.7.3.3 connecteTTN

```
void Communication::connecteTTN ( ) [slot]
```

Références [abonnement](#), [APP\\_TITRE](#), [client](#), [etatClientConnexion\(\)](#), et [nomTopic](#).

Référencé par [Communication\(\)](#).

```
00072 {
00073     qDebug() << Q_FUNC_INFO << nomTopic;
00074     // Le client est maintenant connecté
00075     emit etatClientConnexion(true); // pour l'IHM
00076     // Souscription à un topic :
00077     abonnement = client->subscribe(nomTopic);
00078     if (!abonnement)
00079     {
00080         qDebug() << Q_FUNC_INFO << "Impossible de s'abonner au broker TTN !";
00081         QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Impossible de
s'abonner au broker The Things Network!"));
00082     }
00083 }
```

### 9.7.3.4 deconnecterTTN()

```
void Communication::deconnecterTTN ( )
```

Références [client](#).

Référencé par [~Communication\(\)](#).

```
00063 {
00064     client->disconnectFromHost();
00065 }
```



## 9.7.3.5 deconnecteTTN

```
void Communication::deconnecteTTN ( ) [slot]
```

Références [etatClientConnexion\(\)](#).

Référencé par [Communication\(\)](#).

```
00090 {
00091     qDebug() << Q_FUNC_INFO;
00092     // Le client est maintenant déconnecté
00093     emit etatClientConnexion(false); // pour l'IHM
00094 }
```

## 9.7.3.6 erreurClientTTN

```
void Communication::erreurClientTTN ( ) [slot]
```

**A faire** Faire une boîte de dialogue d'information sur l'erreur rencontrée

Références [client](#).

Référencé par [Communication\(\)](#).

```
00144 {
00145     /*QmqtClient::NoError 0 No error occurred.
00146     QmqtClient::InvalidProtocolVersion 1 The broker does not accept a connection using the specified
00147     protocol version.
00148     QmqtClient::IdRejected 2 The client ID is malformed. This might be related to its
00149     length.
00149     QmqtClient::ServerUnavailable 3 The network connection has been established, but the
00150     service is unavailable on the broker side.
00150     QmqtClient::BadUsernameOrPassword 4 The data in the username or password is malformed.
00151     QmqtClient::NotAuthorized 5 The client is not authorized to connect.
00152     QmqtClient::TransportInvalid 256 The underlying transport caused an error. For example, the
00152     connection might have been interrupted unexpectedly.
00153     QmqtClient::ProtocolViolation 257 The client encountered a protocol violation, and therefore
00153     closed the connection.
00154     QmqtClient::UnknownError 258 An unknown error occurred.
00155     QmqtClient::Mqtt5SpecificError 259 The error is related to MQTT protocol level 5. A reason
00155     code might provide more details.*/
00156
00160     qDebug() << Q_FUNC_INFO << "code erreur" << client->error();
00161 }
```

## 9.7.3.7 etatClientConnexion

```
void Communication::etatClientConnexion (
    bool connexion ) [signal]
```

Référencé par [connecteTTN\(\)](#), et [deconnecteTTN\(\)](#).

## 9.7.3.8 messageRecu

```
void Communication::messageRecu (
    const QByteArray & message,
    const QMqttTopicName & topic ) [signal]
```

Référencé par [receptionner\(\)](#).

## 9.7.3.9 receptionner

```
void Communication::receptionner (
    const QByteArray & message,
    const QMqttTopicName & topic ) [slot]
```

**Paramètres**

<i>message</i>	correspondant au données reçu
<i>topic</i>	correspondant au nom de topic

Références [messageReçu\(\)](#).

Référencé par [Communication\(\)](#).

```
00104 {
00105     //qDebug() << Q_FUNC_INFO << topic;
00106     emit messageReçu(message, topic);
00107 }
```

**9.7.3.10 reponsePingTTN**

```
void Communication::reponsePingTTN ( ) [slot]
```

Référencé par [Communication\(\)](#).

```
00115 {
00116     // Pour les tests (requestPing())
00117     qDebug() << Q_FUNC_INFO;
00118 }
```

**9.7.4 Documentation des données membres****9.7.4.1 abonnement**

```
QMqttSubscription* Communication::abonnement [private]
```

Référencé par [connecteTTN\(\)](#), et [~Communication\(\)](#).

**9.7.4.2 client**

```
QMqttClient* Communication::client [private]
```

Référencé par [changementEtatTTN\(\)](#), [Communication\(\)](#), [connecterTTN\(\)](#), [connecteTTN\(\)](#), [deconnecterTTN\(\)](#), [erreurClientTTN\(\)](#), et [~Communication\(\)](#).

**9.7.4.3 donneesRucheTTN**

```
QStringList Communication::donneesRucheTTN [private]
```

Référencé par [connecterTTN\(\)](#).

## 9.7.4.4 nomTopic

QString Communication::nomTopic [private]

Référencé par [Communication\(\)](#), et [connecteTTN\(\)](#).

## 9.7.4.5 portsTTn

PortsTTN Communication::portsTTn [private]

La documentation de cette classe a été générée à partir des fichiers suivants :

- [communication.h](#)
- [communication.cpp](#)

## 9.8 Référence de la classe fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySql

Classe permettant de se connecter à MySQL en arrière-plan.

Graphe de collaboration de fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySql :

fr.campus.laurainc.honeybee. BaseDeDonnees.ConnexionMySql
- messageConnexion
# doInBackground() # onProgressUpdate() # onPostExecute()

## Fonctions membres protégées

- Boolean [doInBackground](#) (Void... unused)  
*Méthode de connexion qui s'exécute dans un autre thread.*
- void [onProgressUpdate](#) (Integer... progress)  
*Méthode permettant d'indiquer la progression de la tâche d'arrière plan (s'exécute dans le thread UI)*
- void [onPostExecute](#) (Boolean result)

## Attributs privés

- String [messageConnexion](#) = ""

## 9.8.1 Documentation des fonctions membres

### 9.8.1.1 doInBackground()

```
fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySQL.doInBackground (
    Void... unused ) [protected]
```

Références [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#).

```
00300      {
00301          try
00302          {
00303              // chargement du pilote JDBC MySQL
00304              publishProgress(1);
00305              Class.forName("com.mysql.jdbc.Driver").newInstance();
00306          }
00307          catch (Exception e)
00308          {
00309              e.printStackTrace();
00310              Log.e(TAG, "doInBackground -> exception : " + e.toString());
00311              messageConnexion = "Erreur connexion MySQL !";
00312          }
00313          try
00314          {
00315              publishProgress(50);
00316              if (!estConnecte())
00317              {
00318                  connexion = DriverManager.getConnection(url,
00319                      username, password);
00320              }
00321              publishProgress(75);
00322              Log.d(TAG, "doInBackground -> connecte : " + !connexion.isClosed());
00323              if(!connexion.isClosed())
00324                  messageConnexion = "Connexion MySQL réussie !";
00325              publishProgress(100);
00326              return !connexion.isClosed();
00327          }
00328          catch (SQLException e)
00329          {
00330              e.printStackTrace();
00331              Log.e(TAG, "doInBackground -> SQLException : " + e.getMessage());
00332              Log.e(TAG, "doInBackground -> SQLState : " + e.getSQLState());
00333              Log.e(TAG, "doInBackground -> VendorError : " + e.getErrorCode());
00334              messageConnexion = "Erreur connexion MySQL !";
00335              publishProgress(100);
00336              return false;
00337          }
00338      }
```

### 9.8.1.2 onPostExecute()

```
void fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySQL.onPostExecute (
    Boolean result ) [protected]
```

```
00362      {
00363          Log.d(TAG, "onPostExecute -> result : " + result);
00364          Log.d(TAG, "onPostExecute -> message : " + messageConnexion);
00365          if(result)
00366          {
00367              if(activite != null)
00368              {
00369                  // Ici on peut accéder à la partie UI d'une activité
00370                  // et/ou appeler une méthode d'une activité
00371              }
00372          }
00373          else
00374          {
00375              if(activite != null)
00376              {
00377                  // Ici on peut accéder à la partie UI d'une activité
00378                  // et/ou appeler une méthode d'une activité
00379              }
00380          }
00381      }
```

### 9.8.1.3 onProgressUpdate()

```
fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySQL.onProgressUpdate (
    Integer... progress ) [protected]
```

`doInBackground` peut appeler `publishProgress()` pour indiquer l'avancement du traitement ce qui aura pour effet d'appeler automatiquement [onProgressUpdate\(\)](#)

```
00346     {
00347         Log.v(TAG, "onProgressUpdate -> progression : " + progress[0]);
00348         if(activite != null)
00349         {
00350             // Ici on peut accéder à la partie UI d'une activité
00351             // et/ou appeler une méthode d'une activité
00352             //((MainActivity) activite).afficherProgression(progress[0]);
00353         }
00354     }
```

## 9.8.2 Documentation des données membres

### 9.8.2.1 messageConnexion

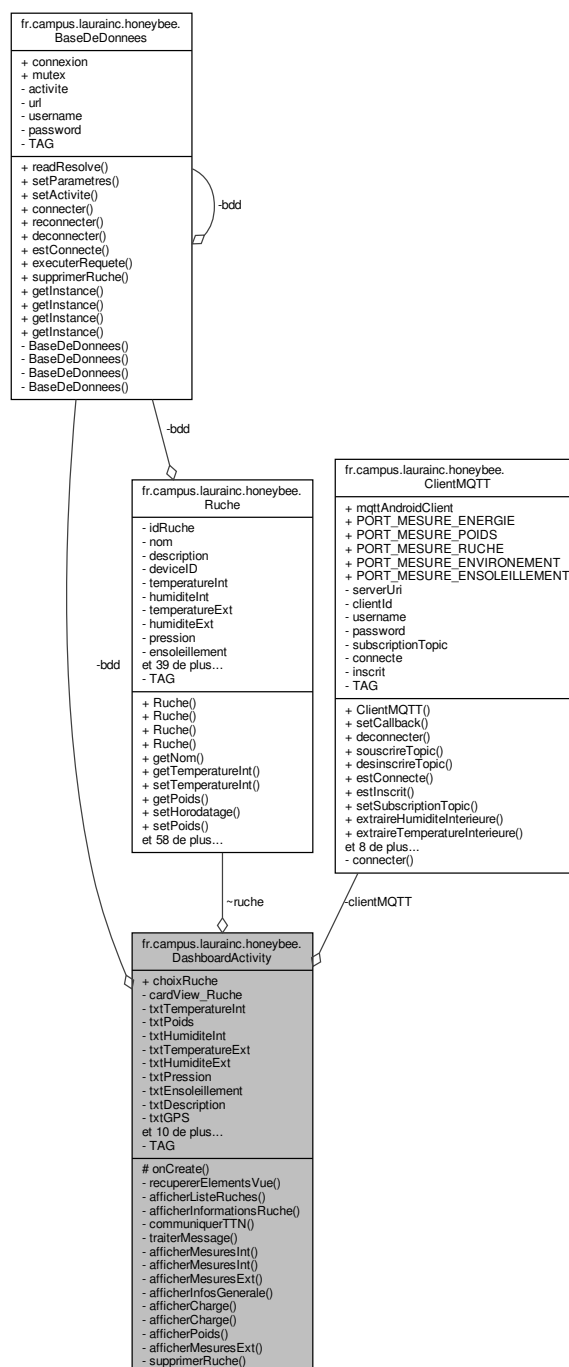
```
String fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySQL.messageConnexion = "" [private]
```

La documentation de cette classe a été générée à partir du fichier suivant :

— [BaseDeDonnees.java](#)

## 9.9 Référence de la classe fr.campus.laurainc.honeybee.DashboardActivity

Graphe de collaboration de fr.campus.laurainc.honeybee.DashboardActivity :



### Attributs publics

— Spinner `choixRuche`

### Fonctions membres protégées

— void `onCreate` (Bundle savedInstanceState)

## Fonctions membres privées

- void [recupererElementsVue](#) ()
- void [afficherListeRuches](#) ()
- void [afficherInformationsRuche](#) ()
- Affichage des informations d'une ruche dans la vue.*
- void [communiquerTTN](#) (String deviceId)
- void [traiterMessage](#) (MqttMessage mqttMessage)
- void [afficherMesuresInt](#) (double temperatureInterieure, double humiditeInterieure, String Horodatage)
- void [afficherMesuresInt](#) (double temperatureInterieure, double humiditeInterieure, double poids)
- void [afficherMesuresExt](#) (double temperatureExterieur, double humiditeExterieur, double pression, double ensoleillement)
- void [afficherInfosGenerale](#) (String description, String longitude, String latitude)
- void [afficherCharge](#) (int charge)
- void [afficherCharge](#) (int charge, String horodatage)
- void [afficherPoids](#) (int Poids, String Horodatage)
- void [afficherMesuresExt](#) (double Humidite, double Temperature, double Pression, String Horodatage)
- void [supprimerRuche](#) ()

## Attributs privés

- [BaseDeDonnees bdd](#) = null
- l'objet permettant un accès à la base de données MySQL*
- [ClientMQTT clientMQTT](#) = null
- [CardView cardView\\_Ruche](#)
- la vue contenant les informations générales de la ruche*
- [TextView txtTemperatureInt](#)
- [TextView txtPoids](#)
- [TextView txtHumiditeInt](#)
- [TextView txtTemperatureExt](#)
- [TextView txtHumiditeExt](#)
- [TextView txtPression](#)
- [TextView txtEnsoleillement](#)
- [TextView txtDescription](#)
- [TextView txtGPS](#)
- [TextView txtCharge](#)
- [TextView txtAlertes](#)
- [ImageView imgCharge](#)
- [ImageView carte](#)
- [Button graphique](#)
- [TextView donneesTTN](#)
- [TextView txtHorodatage](#)
- [FloatingActionButton boutonSupprimerRuche](#)
- [ArrayList< String > mesRuches](#)
- final Handler [handler](#)

## Attributs privés statiques

- static final String [TAG](#) = "DashboardActivity"
- le TAG de la classe pour les logs*

## 9.9.1 Documentation des fonctions membres

9.9.1.1 [afficherCharge\(\)](#) [1/2]

```
void fr.campus.laurainc.honeybee.DashboardActivity.afficherCharge (
    int charge ) [private]
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00296     {
00297         txtCharge.setText (valueOf (charge) + " %");
00298         if (charge > 85)
00299             imgCharge.setImageResource (R.drawable.battery1);
00300         else if (charge <= 85 && charge > 60)
00301             imgCharge.setImageResource (R.drawable.battery);
00302         else if (charge <= 60 && charge > 35)
00303             imgCharge.setImageResource (R.drawable.battery3);
00304         else
00305             imgCharge.setImageResource (R.drawable.battery2);
00306     }
```

### 9.9.1.2 afficherCharge() [2/2]

```
void fr.campus.laurainc.honeybee.DashboardActivity.afficherCharge (
    int charge,
    String horodatage ) [private]

00309 {
00310     txtCharge.setText(valueOf(charge) + " %");
00311     txtHorodatage.setText(valueOf(horodatage));
00312     if (charge > 85)
00313         imgCharge.setImageResource(R.drawable.battery1);
00314     else if (charge <= 85 && charge > 60)
00315         imgCharge.setImageResource(R.drawable.battery);
00316     else if (charge <= 60 && charge > 35)
00317         imgCharge.setImageResource(R.drawable.battery3);
00318     else
00319         imgCharge.setImageResource(R.drawable.battery2);
00320 }
```

### 9.9.1.3 afficherInformationsRuche()

```
fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche ( ) [private]
```

Références [fr.campus.laurainc.honeybee.DashboardActivity.afficherCharge\(\)](#), [fr.campus.laurainc.honeybee.DashboardActivity.afficherInfosGenerale\(\)](#), [fr.campus.laurainc.honeybee.DashboardActivity.afficherMesuresExt\(\)](#), [fr.campus.laurainc.honeybee.DashboardActivity.afficherMesuresInt\(\)](#), [fr.campus.laurainc.honeybee.DashboardActivitycommuniquerTTN\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getCharge\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getDescription\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getDeviceID\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getEnsoleillement\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHumiditeExt\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHumiditeInt\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getLatitude\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getLongitude\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getNom\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPoids\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPression\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getTemperatureExt\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.getTemperatureInt\(\)](#).

```
00159 {
00160     if(ruche != null)
00161     {
00162         Log.d(TAG, "Ruche : " + ruche.getNom());
00163         afficherCharge(ruche.getCharge());
00164         afficherInfosGenerale(ruche.getDescription(), ruche.
getLongitude(), ruche.getLatitude());
00165         afficherMesuresInt(ruche.getTemperatureInt(), ruche.
getHumiditeInt(), ruche.getPoids());
00166         Log.d(TAG, "Affichage des données intérieures");
00167         afficherMesuresExt(ruche.getTemperatureExt(), ruche.
getHumiditeExt(), ruche.getPression(), ruche.
getEnsoleillement());
00168         Log.d(TAG, "Affichage des données extérieures");
00169         donneesTTN.setText("En attente des données du serveur TTN ...");
00170         String topic = "mes_ruches/devices/" + ruche.getDeviceID() + "/up";
00171         clientMQTT = new ClientMQTT(ruche.getDeviceID(), topic,
getApplicationContext());
00172         communiquerTTN(ruche.getDeviceID());
00173     }
00174 }
```

### 9.9.1.4 afficherInfosGenerale()

```
void fr.campus.laurainc.honeybee.DashboardActivity.afficherInfosGenerale (
    String description,
    String longitude,
    String latitude ) [private]
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#).

```
00290 {
00291     txtDescription.setText(description);
00292     txtGPS.setText(longitude + ", " + latitude);
00293 }
```



## 9.9.1.5 afficherListeRuches()

```
void fr.campus.laurainc.honeybee.DashboardActivity.afficherListeRuches ( ) [private]
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.deconnecter\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getListeRuches\(\)](#), [fr.campus.laurainc.honeybee.DashboardActivity.handler](#), et [fr.campus.laurainc.honeybee.DashboardActivity.mesRuches](#).

```
00120         {
00121             mesRuches = ruche.getListeRuches();
00122             ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, R.layout.item_spinner ,
mesRuches);
00123             adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
00124             choixRuche.setAdapter(adapter);
00125             choixRuche.setSelection(0);
00126             //adapter.setNotifyOnChange(true);
00127
00128             choixRuche.setOnItemClickListener(new AdapterView.OnItemClickListener()
00129             {
00130                 @Override
00131                 public void onItemClick(AdapterView<?> arg0, View arg1, int position, long id)
00132                 {
00133                     Toast.makeText(getApplicationContext(), mesRuches.get(position), Toast.LENGTH_SHORT).show
00134                 );
00135                     Log.d(TAG, "position : " + position);
00136                     if(clientMQTT != null)
00137                     {
00138                         if(clientMQTT.estConnecte())
00139                             clientMQTT.deconnecter();
00140                     }
00141                     // Instancie une ruche
00142                     ruche = new Ruche(mesRuches.get(position), handler);
00143                 }
00144
00145                 @Override
00146                 public void onNothingSelected(AdapterView<?> arg0)
00147                 {
00148                     // TODO Auto-generated method stub
00149                 }
00150             });
00151         }
```

## 9.9.1.6 afficherMesuresExt() [1/2]

```
void fr.campus.laurainc.honeybee.DashboardActivity.afficherMesuresExt (
    double temperatureExterieur,
    double humiditeExterieur,
    double pression,
    double ensoleillement ) [private]
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00282     {
00283         txtTemperatureExt.setText(valueOf(temperatureExterieur) + " °C");
00284         txtHumiditeExt.setText(valueOf(humiditeExterieur) + " %");
00285         txtPression.setText(valueOf(pression) + " hPa");
00286         txtEnsoleillement.setText(valueOf(ensoleillement) + " Watt/m²");
00287     }
```

**9.9.1.7 afficherMesuresExt()** [2/2]

```

void fr.campus.laurainc.honeybee.DashboardActivity.afficherMesuresExt (
    double Humidite,
    double Temperature,
    double Pression,
    String Horodatage ) [private]

00330 {
00331     txtHumiditeExt.setText (valueOf (Humidite) + " %");
00332     txtTemperatureExt.setText (valueOf (Temperature) + " °C");
00333     txtPression.setText (valueOf (Pression) + " hPa");
00334     txtHorodatage.setText (Horodatage);
00335 }

```

**9.9.1.8 afficherMesuresInt()** [1/2]

```

void fr.campus.laurainc.honeybee.DashboardActivity.afficherMesuresInt (
    double temperatureInterieure,
    double humiditeInterieure,
    String Horodatage ) [private]

```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```

00268 {
00269     txtTemperatureInt.setText (valueOf (temperatureInterieure) + " °C");
00270     txtHumiditeInt.setText (valueOf (humiditeInterieure) + " %");
00271     txtHorodatage.setText (Horodatage);
00272 }

```

**9.9.1.9 afficherMesuresInt()** [2/2]

```

void fr.campus.laurainc.honeybee.DashboardActivity.afficherMesuresInt (
    double temperatureInterieure,
    double humiditeInterieure,
    double poids ) [private]

00275 {
00276     txtTemperatureInt.setText (valueOf (temperatureInterieure) + " °C");
00277     txtHumiditeInt.setText (valueOf (humiditeInterieure) + " %");
00278     txtPoids.setText (valueOf (poids) + " Kg");
00279 }

```

**9.9.1.10 afficherPoids()**

```

void fr.campus.laurainc.honeybee.DashboardActivity.afficherPoids (
    int Poids,
    String Horodatage ) [private]

```

Références [fr.campus.laurainc.honeybee.ClientMQTT.arrondir\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```

00323 {
00324     double poidsKg = clientMQTT.arrondir (((double) Poids / 1000.), 1);
00325     txtPoids.setText (valueOf (poidsKg) + " Kg");
00326     txtHorodatage.setText (Horodatage);
00327 }

```

## 9.9.1.11 communiquerTTN()

```
void fr.campus.laurainc.honeybee.DashboardActivity.communiqueTTN (
    String deviceId ) [private]
```

Références [fr.campus.laurainc.honeybee.ClientMQTT.setCallback\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#).

```
00179     {
00180         Log.d(TAG, "deviceId : " + deviceId);
00181         clientMQTT.setCallback(new MqttCallbackExtended() {
00182             @Override
00183             public void connectComplete(boolean b, String s) {
00184             }
00185
00186             @Override
00187             public void connectionLost(Throwable throwable) {
00188             }
00189
00190             @Override
00191             public void messageArrived(String topic, MqttMessage mqttMessage) throws Exception {
00192                 Log.d(TAG, "Topic : " + topic);
00193                 Log.d(TAG, "Message : " + mqttMessage.toString());
00194
00195                 // débogage IHM
00196                 donneesTTN.setText(mqttMessage.toString());
00197                 traiterMessage(mqttMessage);
00198             }
00199
00200             @Override
00201             public void deliveryComplete(IMqttDeliveryToken iMqttDeliveryToken) {
00202             }
00203         });
00204     }
00205 }
```

## 9.9.1.12 onCreate()

```
void fr.campus.laurainc.honeybee.DashboardActivity.onCreate (
    Bundle savedInstanceState ) [protected]
```

Références [fr.campus.laurainc.honeybee.DashboardActivity.handler](#), [fr.campus.laurainc.honeybee.DashboardActivity.recupererElementsVue\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.recupererListeRuches\(\)](#).

```
00066     {
00067         super.onCreate(savedInstanceState);
00068         setContentView(R.layout.activity_dash_board);
00069
00070         // Récupère les éléments de la vue d'une Ruche dans l'IHM
00071         recupererElementsVue();
00072
00073         // Instancie une ruche
00074         ruche = new Ruche(handler);
00075         ruche.recupererListeRuches();
00076     }
```

### 9.9.1.13 recupererElementsVue()

```
void fr.campus.laurainc.honeybee.DashboardActivity.recupererElementsVue ( ) [private]
```

Références [fr.campus.laurainc.honeybee.Ruche.getLatitude\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.getLongitude\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.onCreate\(\)](#).

```
00078         {
00079             cardView_Ruche = findViewById(R.id.CardView_Ruche);
00080             txtTemperatureInt = (TextView) this.findViewById(R.id.txt_temp_int);
00081             txtPoids = (TextView) this.findViewById(R.id.txtPoids);
00082             txtHumiditeInt = (TextView) this.findViewById(R.id.txt_humidite_int);
00083             txtTemperatureExt = (TextView) this.findViewById(R.id.txt_temp_ext);
00084             txtHumiditeExt = (TextView) this.findViewById(R.id.txt_humidite_ext);
00085             txtPression = (TextView) this.findViewById(R.id.txt_pression);
00086             txtEnsoleillement = (TextView) findViewById(R.id.txt_ensoleillement);
00087             txtDescription = (TextView) findViewById(R.id.tv_description);
00088             txtGPS = (TextView) findViewById(R.id.tv_GPS);
00089             donneesTTN = (TextView) this.findViewById(R.id.DonneesTTN);
00090             txtHorodatage = (TextView) this.findViewById(R.id.txt_horodatage);
00091             txtCharge = (TextView) this.findViewById(R.id.charge);
00092             choixRuche = (Spinner) findViewById(R.id.choixRuche);
00093             imgCharge = (ImageView) findViewById(R.id.image_charge);
00094             boutonSupprimerRuche = (FloatingActionButton) findViewById(R.id.fb_btnSupprimer
00095         );
00096             carte = (ImageView) findViewById(R.id.btn_carte);
00097
00098             final Intent actualisation = new Intent(DashboardActivity.this, DashboardActivity.class);
00099             boutonSupprimerRuche.setOnClickListener(new View.OnClickListener() {
00100                 @Override
00101                 public void onClick(View v) {
00102                     //supprimerRuche();
00103                     startActivity(actualisation);
00104                 }
00105             });
00106
00107             final Intent map = new Intent(DashboardActivity.this, CarteActivity.class);
00108             carte.setOnClickListener(new View.OnClickListener() {
00109                 @Override
00110                 public void onClick(View v) {
00111                     map.putExtra("longitude", ruche.getLongitude());
00112                     map.putExtra("latitude", ruche.getLatitude());
00113                     Log.d(TAG, "Longitude : " + ruche.getLongitude() + " / Longitude : " + " "
00114 + ruche.getLatitude());
00115                     startActivity(map);
00116                 }
00117             });
00118         }
```

### 9.9.1.14 supprimerRuche()

```
void fr.campus.laurainc.honeybee.DashboardActivity.supprimerRuche ( ) [private]
```

Références [fr.campus.laurainc.honeybee.Ruche.getIdRuche\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.supprimerRuche\(\)](#).

```
00338     {
00339         bdd = BaseDeDonnees.getInstance();
00340         bdd.supprimerRuche(ruche.getIdRuche());
00341     }
```

## 9.9.1.15 traiterMessage()

```
void fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage (
    MqttMessage mqttMessage ) [private]
```

Références [fr.campus.laurainc.honeybee.DashboardActivity.afficherCharge\(\)](#), [fr.campus.laurainc.honeybee.DashboardActivity.afficherMesuresExt\(\)](#), [fr.campus.laurainc.honeybee.DashboardActivity.afficherMesuresInt\(\)](#), [fr.campus.laurainc.honeybee.DashboardActivity.afficherPoids\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireCharge\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireEnsoleillement\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireHorodatage\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireHumiditeExterieur\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireHumiditeInterieur\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extrairePoids\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extrairePression\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireTemperatureExterieur\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.extraireTemperatureInterieur\(\)](#), [fr.campus.laurainc.honeybee.ClientMQTT.PORT\\_MESURE\\_ENERGIE](#), [fr.campus.laurainc.honeybee.ClientMQTT.PORT\\_MESURE\\_ENSOLEILLEMENT](#), [fr.campus.laurainc.honeybee.ClientMQTT.PORT\\_MESURE\\_ENVIRONNEMENT](#), [fr.campus.laurainc.honeybee.ClientMQTT.PORT\\_MESURE\\_POIDS](#), [fr.campus.laurainc.honeybee.ClientMQTT.PORT\\_MESURE\\_RUCHE](#), [fr.campus.laurainc.honeybee.Ruche.setCharge\(\)](#), [fr.campus.laurainc.honeybee.Ruche.setEnsoleillement\(\)](#), [fr.campus.laurainc.honeybee.Ruche.setHorodatage\(\)](#), [fr.campus.laurainc.honeybee.Ruche.setHumiditeExt\(\)](#), [fr.campus.laurainc.honeybee.Ruche.setHumiditeInt\(\)](#), [fr.campus.laurainc.honeybee.Ruche.setPoids\(\)](#), [fr.campus.laurainc.honeybee.Ruche.setPression\(\)](#), [fr.campus.laurainc.honeybee.Ruche.setTemperatureExt\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.setTemperatureInt\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivitycommuniquerTTN\(\)](#).

```
00210    {
00211        final String message = mqttMessage.toString();
00212
00213        try
00214        {
00215            JSONObject jsonObjet = new JSONObject(message);
00216            int port = jsonObjet.getInt("port");
00217            String Horodatage = clientMQTT.extraireHorodatage(mqttMessage);
00218            Log.d(TAG, "Port : " + port);
00219            switch(port)
00220            {
00221                case ClientMQTT.PORT_MESURE_POIDS:
00222                    int poids = clientMQTT.extrairePoids(mqttMessage); // poids en
00223                    grammes
00224                    afficherPoids(poids, Horodatage);
00225                    ruche.setPoids(poids);
00226                    break;
00227                case ClientMQTT.PORT_MESURE_RUCHE:
00228                    double temperatureInterieur = clientMQTT.
00229                    extraireTemperatureInterieur(mqttMessage);
00230                    double humiditeInterieur = clientMQTT.
00231                    extraireHumiditeInterieur(mqttMessage);
00232                    afficherMesuresInt(temperatureInterieur, humiditeInterieur,
00233                    Horodatage);
00234                    ruche.setTemperatureInt(temperatureInterieur);
00235                    ruche.setHumiditeInt(temperatureInterieur);
00236                    break;
00237                case ClientMQTT.PORT_MESURE_ENERGIE:
00238                    int charge = clientMQTT.extraireCharge(mqttMessage);
00239                    afficherCharge(charge, Horodatage);
00240                    ruche.setCharge(charge);
00241                    break;
00242                case ClientMQTT.PORT_MESURE_ENVIRONNEMENT:
00243                    double temperatureExterieur = clientMQTT.
00244                    extraireTemperatureExterieur(mqttMessage);
00245                    double humiditeExterieur = clientMQTT.
00246                    extraireHumiditeExterieur(mqttMessage);
00247                    double pression = clientMQTT.extrairePression(mqttMessage);
00248                    afficherMesuresExt(humiditeExterieur, temperatureExterieur,
00249                    pression, Horodatage);
00250                    ruche.setTemperatureExt(temperatureExterieur);
00251                    ruche.setHumiditeExt(humiditeExterieur);
00252                    ruche.setPression(pression);
00253                    break;
00254                case ClientMQTT.PORT_MESURE_ENSOLEILLEMENT:
00255                    double ensoleillement = clientMQTT.
00256                    extraireEnsoleillement(mqttMessage);
00257                    txtEnsoleillement.setText(valueOf(ensoleillement) + " Watt/m²");
00258                    ruche.setEnsoleillement(ensoleillement);
00259            }
00260        }
00261    }
```

```
00259         ruche.setHorodatage(Horodatage);
00260     }
00261     catch (JSONException e)
00262     {
00263         e.printStackTrace();
00264     }
00265 }
```

## 9.9.2 Documentation des données membres

### 9.9.2.1 bdd

`BaseDeDonnees` `fr.campus.laurainc.honeybee.DashboardActivity.bdd = null` [private]

### 9.9.2.2 boutonSupprimerRuche

`FloatingActionButton` `fr.campus.laurainc.honeybee.DashboardActivity.boutonSupprimerRuche` [private]

### 9.9.2.3 cardView\_Ruche

`CardView` `fr.campus.laurainc.honeybee.DashboardActivity.cardView_Ruche` [private]

### 9.9.2.4 carte

`ImageView` `fr.campus.laurainc.honeybee.DashboardActivity.carte` [private]

### 9.9.2.5 choixRuche

`Spinner` `fr.campus.laurainc.honeybee.DashboardActivity.choixRuche`

### 9.9.2.6 clientMQTT

`ClientMQTT` `fr.campus.laurainc.honeybee.DashboardActivity.clientMQTT = null` [private]

### 9.9.2.7 donneesTTN

`TextView` `fr.campus.laurainc.honeybee.DashboardActivity.donneesTTN` [private]

### 9.9.2.8 graphique

Button fr.campus.laurainc.honeybee.DashboardActivity.graphique [private]

### 9.9.2.9 handler

final Handler fr.campus.laurainc.honeybee.DashboardActivity.handler [private]

**Valeur initiale :**

```
= new Handler()
{
    public void handleMessage(Message msg)
    {
        super.handleMessage(msg);
        switch (msg.what)
        {
            case HoneyBee.REQUETE_SQL_ERREUR:
                Log.d(TAG, "handleMessage -> REQUETE SQL ERREUR");
                break;
            case HoneyBee.REQUETE_SQL_OK:
                Log.d(TAG, "handleMessage -> REQUETE SQL OK");
                break;
            case HoneyBee.REQUETE_SQL_LISTE_RUCHES:
                Log.d(TAG, "handleMessage -> REQUETE SQL LISTE RUCHES");
                afficherListeRuches();
                break;
            case HoneyBee.REQUETE_SQL_RUCHE:
                Log.d(TAG, "handleMessage -> REQUETE SQL RUCHE");
                afficherInformationsRuche();
                break;
        }
    }
}
```

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherListeRuches\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.onCreate\(\)](#).

### 9.9.2.10 imgCharge

ImageView fr.campus.laurainc.honeybee.DashboardActivity.imgCharge [private]

### 9.9.2.11 mesRuches

ArrayList<String> fr.campus.laurainc.honeybee.DashboardActivity.mesRuches [private]

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherListeRuches\(\)](#).

### 9.9.2.12 TAG

final String fr.campus.laurainc.honeybee.DashboardActivity.TAG = "DashboardActivity" [static], [private]

**9.9.2.13 txtAlertes**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtAlertes [private]

**9.9.2.14 txtCharge**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtCharge [private]

**9.9.2.15 txtDescription**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtDescription [private]

**9.9.2.16 txtEnsoleillement**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtEnsoleillement [private]

**9.9.2.17 txtGPS**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtGPS [private]

**9.9.2.18 txtHorodatage**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtHorodatage [private]

**9.9.2.19 txtHumiditeExt**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtHumiditeExt [private]

**9.9.2.20 txtHumiditeInt**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtHumiditeInt [private]

**9.9.2.21 txtPoids**

TextView fr.campus.laurainc.honeybee.DashboardActivity.txtPoids [private]



## 9.9.2.22 txtPression

```
TextView fr.campus.laurainc.honeybee.DashboardActivity.txtPression [private]
```

## 9.9.2.23 txtTemperatureExt

```
TextView fr.campus.laurainc.honeybee.DashboardActivity.txtTemperatureExt [private]
```

## 9.9.2.24 txtTemperatureInt

```
TextView fr.campus.laurainc.honeybee.DashboardActivity.txtTemperatureInt [private]
```

La documentation de cette classe a été générée à partir du fichier suivant :

— [DashboardActivity.java](#)

## 9.10 Référence de la classe fr.campus.laurainc.honeybee.DetailsRucheActivity

Graphe de collaboration de fr.campus.laurainc.honeybee.DetailsRucheActivity :

fr.campus.laurainc.honeybee. DetailsRucheActivity
~ idRuche
+ setIdRuche() # onCreate()

## Fonctions membres publiques

— void [setIdRuche](#) (int idRuche)

## Fonctions membres protégées

— void [onCreate](#) (Bundle savedInstanceState)

## 9.10.1 Description détaillée

**A faire** Renommer cette classe

## 9.10.2 Documentation des fonctions membres

### 9.10.2.1 onCreate()

```
void fr.campus.laurainc.honeybee.DetailsRucheActivity.onCreate (
    Bundle savedInstanceState ) [protected]
```

```
00017    {
00018        super.onCreate(savedInstanceState);
00019        setContentView(R.layout.activity_details);
00020    }
00021
00022 }
```

### 9.10.2.2 setIdRuche()

```
void fr.campus.laurainc.honeybee.DetailsRucheActivity.setIdRuche (
    int idRuche )
```

```
00024    {
00025        this.idRuche = idRuche;
00026    }
```

La documentation de cette classe a été générée à partir du fichier suivant :

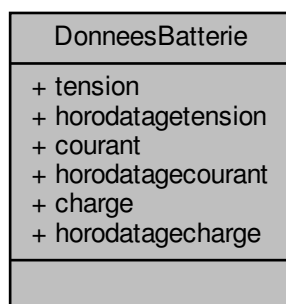
— [DetailsRucheActivity.java](#)

## 9.11 Référence de la structure DonneesBatterie

structure de données pour les mesures de la batterie

```
#include <ruche.h>
```

Graphe de collaboration de DonneesBatterie :



### Attributs publics

- QString [tension](#)
- QString [horodatagetension](#)
- QString [courant](#)
- QString [horodatagecourant](#)
- QString [charge](#)
- QString [horodatagecharge](#)

### 9.11.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

### 9.11.2 Documentation des données membres

#### 9.11.2.1 `charge`

`QString DonneesBatterie::charge`

Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insererDonneesPortBatterie\(\)](#).

#### 9.11.2.2 `courant`

`QString DonneesBatterie::courant`

Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insererDonneesPortBatterie\(\)](#).

#### 9.11.2.3 `horodatagecharge`

`QString DonneesBatterie::horodatagecharge`

Référencé par [Ruche : :extraireDonnees\(\)](#).

#### 9.11.2.4 `horodatagecourant`

`QString DonneesBatterie::horodatagecourant`

Référencé par [Ruche : :extraireDonnees\(\)](#).

#### 9.11.2.5 `horodatagetension`

`QString DonneesBatterie::horodatagetension`

Référencé par [Ruche : :extraireDonnees\(\)](#).

### 9.11.2.6 tension

```
QString DonneesBatterie::tension
```

Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insérerDonneesPortBatterie\(\)](#).

La documentation de cette structure a été générée à partir du fichier suivant :

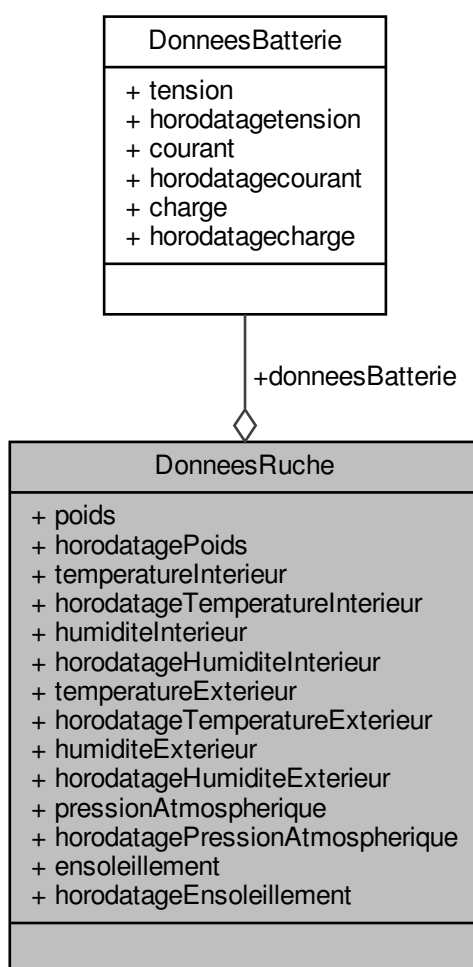
— [ruce.h](#)

## 9.12 Référence de la structure DonneesRuche

structure de données pour les mesures horodatées

```
#include <ruce.h>
```

Graphe de collaboration de DonneesRuche :



**Attributs publics**

- `DonneesBatterie` `donneesBatterie`
- `QString` `poids`
- `QString` `horodatagePoids`
- `QString` `temperatureInterieur`
- `QString` `horodatageTemperatureInterieur`
- `QString` `humiditeInterieur`
- `QString` `horodatageHumiditeInterieur`
- `QString` `temperatureExterieur`
- `QString` `horodatageTemperatureExterieur`
- `QString` `humiditeExterieur`
- `QString` `horodatageHumiditeExterieur`
- `QString` `pressionAtmospherique`
- `QString` `horodatagePressionAtmospherique`
- `QString` `ensoleillement`
- `QString` `horodatageEnsoleillement`

**9.12.1 Description détaillée****Auteur**

Florentin Mellah, Enzo Rossi

**Version**

1.1

**9.12.2 Documentation des données membres****9.12.2.1 `donneesBatterie`**

`DonneesBatterie` `DonneesRuche::donneesBatterie`

Référencé par `Ruche : :extraireDonnees()`, et `Ruche : :insérerDonneesPortBatterie()`.

**9.12.2.2 `ensoleillement`**

`QString` `DonneesRuche::ensoleillement`

Référencé par `Ruche : :extraireDonnees()`, et `Ruche : :insérerDonneesPortEnsoleillement()`.

**9.12.2.3 `horodatageEnsoleillement`**

`QString` `DonneesRuche::horodatageEnsoleillement`

Référencé par `Ruche : :extraireDonnees()`.

**9.12.2.4 horodatageHumiditeExterieur**

`QString DonneesRuche::horodatageHumiditeExterieur`

Référencé par [Ruche : :extraireDonnees\(\)](#).

**9.12.2.5 horodatageHumiditeInterieur**

`QString DonneesRuche::horodatageHumiditeInterieur`

Référencé par [Ruche : :extraireDonnees\(\)](#).

**9.12.2.6 horodatagePoids**

`QString DonneesRuche::horodatagePoids`

Référencé par [Ruche : :extraireDonnees\(\)](#).

**9.12.2.7 horodatagePressionAtmospherique**

`QString DonneesRuche::horodatagePressionAtmospherique`

Référencé par [Ruche : :extraireDonnees\(\)](#).

**9.12.2.8 horodatageTemperatureExterieur**

`QString DonneesRuche::horodatageTemperatureExterieur`

Référencé par [Ruche : :extraireDonnees\(\)](#).

**9.12.2.9 horodatageTemperatureInterieur**

`QString DonneesRuche::horodatageTemperatureInterieur`

Référencé par [Ruche : :extraireDonnees\(\)](#).

**9.12.2.10 humiditeExterieur**

`QString DonneesRuche::humiditeExterieur`

Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insererDonneesPortMesureEnvironnement\(\)](#).

#### 9.12.2.11 humiditeInterieur

QString DonneesRuche::humiditeInterieur

Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insererDonneesPortMesureRuche\(\)](#).

#### 9.12.2.12 poids

QString DonneesRuche::poids

Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insererDonneesPortPoids\(\)](#).

#### 9.12.2.13 pressionAtmospherique

QString DonneesRuche::pressionAtmospherique

Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insererDonneesPortMesureEnvironnement\(\)](#).

#### 9.12.2.14 temperatureExterieur

QString DonneesRuche::temperatureExterieur

Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insererDonneesPortMesureEnvironnement\(\)](#).

#### 9.12.2.15 temperatureInterieur

QString DonneesRuche::temperatureInterieur

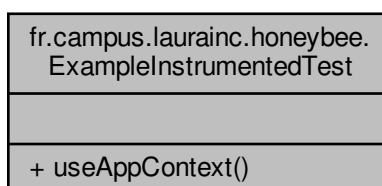
Référencé par [Ruche : :extraireDonnees\(\)](#), et [Ruche : :insererDonneesPortMesureRuche\(\)](#).

La documentation de cette structure a été générée à partir du fichier suivant :

— [ruce.h](#)

### 9.13 Référence de la classe fr.campus.laurainc.honeybee.ExampleInstrumentedTest

Graphe de collaboration de fr.campus.laurainc.honeybee.ExampleInstrumentedTest :



## Fonctions membres publiques

— void [useAppContext](#) ()

## 9.13.1 Description détaillée

Instrumented test, which will execute on an Android device.

## Voir également

[Testing documentation](#)

## 9.13.2 Documentation des fonctions membres

## 9.13.2.1 useAppContext()

```
void fr.campus.laurainc.honeybee.ExampleInstrumentedTest.useAppContext ( )
```

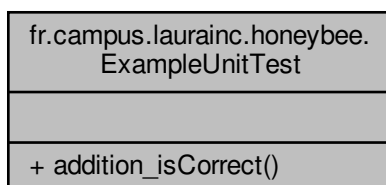
```
00020                                     {
00021         // Context of the app under test.
00022         Context appContext = InstrumentationRegistry.getTargetContext();
00023
00024         assertEquals("fr.campus.laurainc.honeybee", appContext.getPackageName());
00025     }
```

La documentation de cette classe a été générée à partir du fichier suivant :

— [ExampleInstrumentedTest.java](#)

## 9.14 Référence de la classe fr.campus.laurainc.honeybee.ExampleUnitTest

Graphe de collaboration de fr.campus.laurainc.honeybee.ExampleUnitTest :



## Fonctions membres publiques

— void [addition\\_isCorrect](#) ()



### 9.14.1 Description détaillée

Example local unit test, which will execute on the development machine (host).

#### Voir également

[Testing documentation](#)

### 9.14.2 Documentation des fonctions membres

#### 9.14.2.1 addition\_isCorrect()

```
void fr.campus.laurainc.honeybee.ExampleUnitTest.addition_isCorrect ( )
```

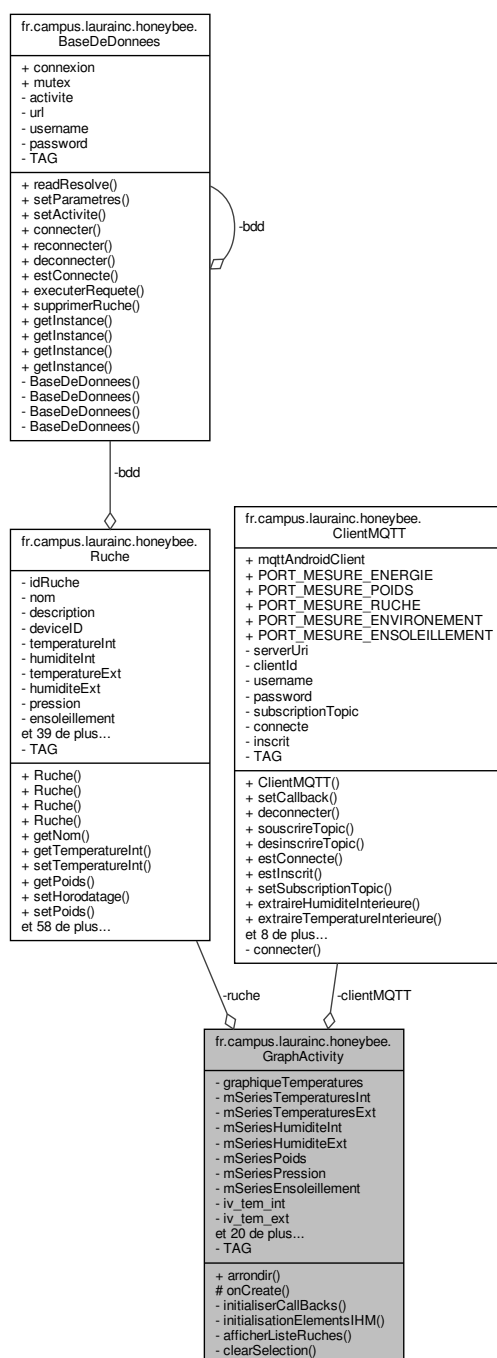
```
00014      {  
00015      assertEquals(4, 2 + 2);  
00016  }
```

La documentation de cette classe a été générée à partir du fichier suivant :

— [ExampleUnitTest.java](#)

## 9.15 Référence de la classe fr.campus.laurainc.honeybee.GraphActivity

Graphe de collaboration de fr.campus.laurainc.honeybee.GraphActivity :



### Fonctions membres publiques

— double [arrondir](#) (double nombre, double nbApresVirgule)

### Fonctions membres protégées

— void [onCreate](#) (Bundle savedInstanceState)

## Fonctions membres privées

- void [initialiserCallbacks](#) ()
- void [initialisationElementsIHM](#) ()
- void [afficherListeRuches](#) ()
- void [clearSelection](#) ()

## Attributs privés

- [Ruche](#) [ruche](#)
- [GraphView](#) [graphiqueTemperatures](#)
- [LineGraphSeries](#)< [DataPoint](#) > [mSeriesTemperaturesInt](#)
- [LineGraphSeries](#)< [DataPoint](#) > [mSeriesTemperaturesExt](#)
- [LineGraphSeries](#)< [DataPoint](#) > [mSeriesHumiditeInt](#)
- [LineGraphSeries](#)< [DataPoint](#) > [mSeriesHumiditeExt](#)
- [LineGraphSeries](#)< [DataPoint](#) > [mSeriesPoids](#)
- [LineGraphSeries](#)< [DataPoint](#) > [mSeriesPression](#)
- [LineGraphSeries](#)< [DataPoint](#) > [mSeriesEnsoleillement](#)
- [Button](#) [iv\\_tem\\_int](#)
- [Button](#) [iv\\_tem\\_ext](#)
- [Button](#) [iv\\_humidite\\_int](#)
- [Button](#) [iv\\_humidite\\_ext](#)
- [Button](#) [iv\\_poids](#)
- [Button](#) [iv\\_ensoleillement](#)
- [Button](#) [iv\\_pression](#)
- [boolean](#) [temp\\_int\\_afficher](#)
- [boolean](#) [temp\\_ext\\_afficher](#)
- [boolean](#) [humidite\\_int\\_afficher](#)
- [boolean](#) [humidite\\_ext\\_afficher](#)
- [boolean](#) [poids\\_afficher](#)
- [boolean](#) [ensoleillement\\_afficher](#)
- [boolean](#) [pression\\_afficher](#)
- [boolean](#) [temps\\_reel\\_afficher](#)
- [ClientMQTT](#) [clientMQTT](#)
- [TextView](#) [tv\\_valeurCourante](#)
- [TextView](#) [tv\\_donneesBas](#)
- [TextView](#) [tv\\_donneesMoyenne](#)
- [TextView](#) [tv\\_donneesHaut](#)
- [Spinner](#) [choixRuche](#)
- [ArrayList](#)< [String](#) > [mesRuches](#)
- [final](#) [Handler](#) [handler](#)

## Attributs privés statiques

- [static final](#) [String](#) [TAG](#) = "GraphActivity"

## 9.15.1 Documentation des fonctions membres

## 9.15.1.1 afficherListeRuches()

```
void fr.campus.laurainc.honeybee.GraphActivity.afficherListeRuches ( ) [private]
```

Références [fr.campus.laurainc.honeybee.Ruche.getListeRuches\(\)](#), [fr.campus.laurainc.honeybee.GraphActivity.handler](#), et [fr.campus.laurainc.honeybee.GraphActivity.mesRuches](#).

```
00347         {
00348             mesRuches = ruche.getListeRuches();
00349             ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, R.layout.item_spinner ,
mesRuches);
00350             adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
00351             choixRuche.setAdapter(adapter);
00352             choixRuche.setSelection(0);
00353             //adapter.setNotifyOnChange(true);
00354
00355             choixRuche.setOnItemClickListener(new AdapterView.OnItemClickListener()
```

```

00356     {
00357         @Override
00358         public void onItemClick(AdapterView<?> arg0, View arg1, int position, long id)
00359         {
00360             Toast.makeText(getBaseContext(), mesRuches.get(position), Toast.LENGTH_SHORT).show
00361         );
00362             Log.d(TAG, "position : " + position);
00363             /*if(clientMQTT != null)
00364             {
00365                 if(clientMQTT.estConnecte())
00366                     clientMQTT.deconnecter();
00367             }*/
00368             // Instancie une ruche
00369             ruche = new Ruche(mesRuches.get(position),
00370 handler);
00371         }
00372         @Override
00373         public void onNothingSelected(AdapterView<?> arg0)
00374         {
00375             // TODO Auto-generated method stub
00376         }
00377     });
00378 }

```

#### 9.15.1.2 arrondir()

```

double fr.campus.laurainc.honeybee.GraphActivity.arrondir (
    double nombre,
    double nbApresVirgule )

00408 {
00409     return(double)((int)(nombre * Math.pow(10,nbApresVirgule) + .5)) / Math.pow(10,nbApresVirgule);
00410 }

```

#### 9.15.1.3 clearSelection()

```
void fr.campus.laurainc.honeybee.GraphActivity.clearSelection ( ) [private]
```

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```

00381 {
00382     iv_tem_ext.setBackgroundResource(0);
00383     temp_ext_afficher = false;
00384
00385     iv_tem_int.setBackgroundResource(0);
00386     temp_int_afficher = false;
00387
00388     iv_humidite_int.setBackgroundResource(0);
00389     humidite_int_afficher = false;
00390
00391     iv_humidite_ext.setBackgroundResource(0);
00392     humidite_ext_afficher = false;
00393
00394     iv_poids.setBackgroundResource(0);
00395     poids_afficher = false;
00396
00397     iv_pression.setBackgroundResource(0);
00398     pression_afficher = false;
00399
00400     iv_enseillement.setBackgroundResource(0);
00401     enseillement_afficher = false;
00402
00403     graphiqueTemperatures.removeAllSeries();
00404
00405 }

```

## 9.15.1.4 initialisationElementsIHM()

```
void fr.campus.laurainc.honeybee.GraphActivity.initialisationElementsIHM ( ) [private]
```

Références [fr.campus.laurainc.honeybee.Ruche.getmSeriesEnsoleillement\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getmSeriesHumiditeExt\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getmSeriesHumiditeInt\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getmSeriesPoids\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getmSeriesPression\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getmSeriesTemperaturesExt\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.getmSeriesTemperaturesInt\(\)](#).

```
00302 {
00303     iv_tem_int = findViewById(R.id.btn_temp_int);
00304     iv_tem_ext = findViewById(R.id.btn_temp_ext);
00305     iv_humidite_int = findViewById(R.id.btn_humidite_int);
00306     iv_humidite_ext = findViewById(R.id.btn_humidite_ext);
00307     iv_poids = findViewById(R.id.btn_poids);
00308     iv_ensoleillement = findViewById(R.id.btn_ensoleillement);
00309     iv_pression = findViewById(R.id.btn_pression);
00310     tv_valeurCourante = findViewById(R.id.tv_valeurCourante);
00311     tv_donneesBas = findViewById(R.id.tv_donneesBas);
00312     tv_donneesMoyenne = findViewById(R.id.tv_donneesMoyenne);
00313     tv_donneesHaut = findViewById(R.id.tv_donneesHaut);
00314     //iv_temps_reel = findViewById(R.id.iv_temp_reel);
00315
00316     temp_int_afficher = true;
00317     temp_ext_afficher = false;
00318     humidite_int_afficher = false;
00319     humidite_ext_afficher = false;
00320     poids_afficher = false;
00321     ensoleillement_afficher = false;
00322     pression_afficher = false;
00323     temps_reel_afficher = false;
00324
00325     mSeriesTemperaturesInt = ruche.
getmSeriesTemperaturesInt();
00326     graphiqueTemperatures.getViewPort().setMaxY(40);
00327     graphiqueTemperatures.getGridLabelRenderer().setNumVerticalLabels(5);
00328     graphiqueTemperatures.removeAllSeries();
00329     graphiqueTemperatures.addSeries(
mSeriesTemperaturesInt);
00330     mSeriesTemperaturesExt = ruche.
getmSeriesTemperaturesExt();
00331     mSeriesHumiditeInt = ruche.getmSeriesHumiditeInt();
00332     mSeriesHumiditeExt = ruche.getmSeriesHumiditeExt();
00333     mSeriesPoids = ruche.getmSeriesPoids();
00334     mSeriesEnsoleillement = ruche.
getmSeriesEnsoleillement();
00335     mSeriesPression = ruche.getmSeriesPression();
00336
00337     mSeriesTemperaturesInt.setColor(Color.RED);
00338     mSeriesHumiditeExt.setColor(Color.RED);
00339     mSeriesHumiditeInt.setColor(Color.BLUE);
00340     mSeriesHumiditeExt.setColor(Color.BLUE);
00341     mSeriesPoids.setColor(Color.GREEN);
00342     mSeriesPression.setColor(Color.GRAY);
00343     mSeriesEnsoleillement.setColor(Color.BLACK);
00344     Log.d(TAG, "Initialisation des éléments de l'IHM");
00345 }
```

## 9.15.1.5 initialiserCallbacks()

```
void fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks ( ) [private]
```

Références [fr.campus.laurainc.honeybee.GraphActivity.clearSelection\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getEnsoleillement\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getEnsoleillement\\_Basse\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getEnsoleillement\\_Haute\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getEnsoleillement\\_Moyenne\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHum\\_ext\\_Basse\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHum\\_ext\\_Haute\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHum\\_ext\\_Moyenne\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHum\\_int\\_Basse\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHum\\_int\\_Haute\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHum\\_int\\_Moyenne\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHumiditeExt\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getHumiditeInt\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPoids\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPoids\\_Basse\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPoids\\_Haute\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPoids\\_Moyenne\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPression\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPression\\_Basse\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPression\\_Haute\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getPression\\_Moyenne\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getTemp\\_ext\\_Basse\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getTemp\\_ext\\_Haute\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getTemp\\_ext\\_Moyenne\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getTemp\\_int\\_Basse\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getTemp\\_int\\_Haute\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getTemp\\_int\\_Moyenne\(\)](#), [fr.campus.laurainc.honeybee.Ruche.getTemperatureExt\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.getTemperatureInt\(\)](#).

```

00124     {
00125         tv_valeurCourante.setText (valueOf (ruche.
getTemperatureInt () ) + "°C");
00126         tv_donneesBas.setText (valueOf (ruche.getTemp_int_Basse () ) + "°C")
;
00127         tv_donneesHaut.setText (valueOf (ruche.
getTemp_int_Haute () ) + "°C");
00128         tv_donneesMoyenne.setText (String.format ("% .2f °C",
ruche.getTemp_int_Moyenne () ));
00129
00130         iv_tem_int.setOnClickListener (new View.OnClickListener () {
00131             @Override
00132             public void onClick (View v) {
00133                 if (temp_int_afficher)
00134                 {
00135                     clearSelection ();
00136                 }
00137                 else
00138                 {
00139                     clearSelection ();
00140                     graphiqueTemperatures.getViewPort ().setMaxY (40);
00141                     graphiqueTemperatures.getViewPort ().setMinY (0);
00142                     graphiqueTemperatures.getGridLabelRenderer ().setNumVerticalLabels (
5);
00143                     iv_tem_int.setBackgroundResource (R.color.BleuSelection);
00144                     tv_valeurCourante.setText (valueOf (ruche.
getTemperatureInt () ) + "°C");
00145                     tv_donneesBas.setText (valueOf (ruche.
getTemp_int_Basse () ) + "°C");
00146                     tv_donneesHaut.setText (valueOf (ruche.
getTemp_int_Haute () ) + "°C");
00147                     tv_donneesMoyenne.setText (String.format ("% .2f °C",
ruche.getTemp_int_Moyenne () ));
00148                     temp_int_afficher = true;
00149                     graphiqueTemperatures.addSeries (
mSeriesTemperaturesInt);
00150
00151                 }
00152             }
00153         });
00154
00155         iv_tem_ext.setOnClickListener (new View.OnClickListener () {
00156             @Override
00157             public void onClick (View v) {
00158                 if (temp_ext_afficher)
00159                 {
00160                     clearSelection ();
00161                 }
00162                 else
00163                 {
00164                     clearSelection ();
00165                     graphiqueTemperatures.getViewPort ().setMaxY (40);
00166                     graphiqueTemperatures.getViewPort ().setMinY (0);
00167                     graphiqueTemperatures.getGridLabelRenderer ().setNumVerticalLabels (
5);
00168                     iv_tem_ext.setBackgroundResource (R.color.BleuSelection);
00169                     tv_valeurCourante.setText (valueOf (ruche.
getTemperatureExt () ) + "°C");
00170                     tv_donneesBas.setText (valueOf (ruche.
getTemp_ext_Basse () ) + "°C");
00171                     tv_donneesHaut.setText (valueOf (ruche.
getTemp_ext_Haute () ) + "°C");
00172                     tv_donneesMoyenne.setText (String.format ("% .2f °C",
ruche.getTemp_ext_Moyenne () ));
00173                     temp_ext_afficher = true;
00174                     graphiqueTemperatures.addSeries (
mSeriesTemperaturesExt);
00175
00176                 }
00177             }
00178         });
00179
00180         iv_humidite_int.setOnClickListener (new View.OnClickListener () {
00181             @Override
00182             public void onClick (View v) {
00183                 if (humidite_int_afficher)
00184                 {
00185                     clearSelection ();
00186                 }
00187                 else
00188                 {
00189                     clearSelection ();
00190                     graphiqueTemperatures.getViewPort ().setMaxY (100);
00191                     graphiqueTemperatures.getViewPort ().setMinY (0);
00192                     graphiqueTemperatures.getGridLabelRenderer ().setNumVerticalLabels (
6);
00193                     iv_humidite_int.setBackgroundResource (R.color.BleuSelection);
00194                     tv_valeurCourante.setText (valueOf (ruche.
getHumiditeInt () ) + "%");
00195                     tv_donneesBas.setText (valueOf (ruche.
getHum_int_Basse () ) + "%");
00196                     tv_donneesHaut.setText (valueOf (ruche.

```

```

    getHum_int_Haute()) + "%");
00196         tv_donneesMoyenne.setText(String.format("%.2f ",
    ruche.getHum_int_Moyenne()) + "%");
00197         humidite_int_afficher = true;
00198         graphiqueTemperatures.addSeries(
    mSeriesHumiditeInt);
00199     }
00200 }
00201 });
00202
00203     iv_humidite_ext.setOnClickListener(new View.OnClickListener() {
00204         @Override
00205         public void onClick(View v) {
00206             if(humidite_ext_afficher)
00207             {
00208                 clearSelection();
00209             }
00210             else
00211             {
00212                 clearSelection();
00213                 graphiqueTemperatures.getViewPort().setMaxY(100);
00214                 graphiqueTemperatures.getViewPort().setMinY(0);
00215                 graphiqueTemperatures.getGridLabelRenderer().setNumVerticalLabels(
    6);
00216                 iv_humidite_ext.setBackgroundResource(R.color.BleuSelection);
00217                 tv_valeurCourante.setText(valueOf(ruche.
    getHumiditeExt()) + "%");
00218                 tv_donneesBas.setText(valueOf(ruche.
    getHum_ext_Basse()) + "%");
00219                 tv_donneesHaut.setText(valueOf(ruche.
    getHum_ext_Haute()) + "%");
00220                 tv_donneesMoyenne.setText(String.format("%.2f ",
    ruche.getHum_ext_Moyenne()) + "%");
00221                 humidite_ext_afficher = true;
00222                 graphiqueTemperatures.addSeries(
    mSeriesHumiditeExt);
00223             }
00224         }
00225     });
00226
00227     iv_poids.setOnClickListener(new View.OnClickListener() {
00228         @Override
00229         public void onClick(View v) {
00230             if(poids_afficher)
00231             {
00232                 clearSelection();
00233             }
00234             else
00235             {
00236                 clearSelection();
00237                 graphiqueTemperatures.getViewPort().setMaxY(80);
00238                 graphiqueTemperatures.getViewPort().setMinY(0);
00239                 graphiqueTemperatures.getGridLabelRenderer().setNumVerticalLabels(
    9);
00240                 iv_poids.setBackgroundResource(R.color.BleuSelection);
00241                 tv_valeurCourante.setText(valueOf(ruche.
    getPoids()) + "Kg");
00242                 tv_donneesBas.setText(valueOf(ruche.
    getPoids_Basse()) + "Kg");
00243                 tv_donneesHaut.setText(valueOf(ruche.
    getPoids_Haute()) + "Kg");
00244                 tv_donneesMoyenne.setText(String.format("%.2f Kg",
    ruche.getPoids_Moyenne()/1000));
00245                 poids_afficher = true;
00246                 graphiqueTemperatures.addSeries(
    mSeriesPoids);
00247             }
00248         }
00249     });
00250
00251     iv_pression.setOnClickListener(new View.OnClickListener() {
00252         @Override
00253         public void onClick(View v) {
00254             if(pression_afficher)
00255             {
00256                 clearSelection();
00257             }
00258             else
00259             {
00260                 clearSelection();
00261                 graphiqueTemperatures.getViewPort().setMaxY(1100);
00262                 graphiqueTemperatures.getViewPort().setMinY(900);
00263                 graphiqueTemperatures.getGridLabelRenderer().setNumVerticalLabels(
    5);
00264                 iv_pression.setBackgroundResource(R.color.BleuSelection);
00265                 tv_valeurCourante.setText(valueOf(ruche.
    getPression()) + "hPa");
00266                 tv_donneesBas.setText(valueOf(ruche.
    getPression_Basse()) + "hPa");
00267                 tv_donneesHaut.setText(valueOf(ruche.
    getPression_Haute()) + "hPa");

```

```

00268         tv_donneesMoyenne.setText(String.format("%.2f hPa",
ruche.getPression_Moyenne()));
00269         pression_afficher = true;
00270         graphiqueTemperatures.addSeries(
mSeriesPression);
00271     }
00272 }
00273 });
00274
00275 iv_enseleillement.setOnClickListener(new View.OnClickListener() {
00276     @Override
00277     public void onClick(View v) {
00278         if(enseleillement_afficher)
00279         {
00280             clearSelection();
00281         }
00282         else
00283         {
00284             clearSelection();
00285             graphiqueTemperatures.getViewPort().setMaxY(1000);
00286             graphiqueTemperatures.getViewPort().setMinY(0);
00287             graphiqueTemperatures.getGridLabelRenderer().setNumVerticalLabels(
6);
00288             iv_enseleillement.setBackgroundResource(R.color.BleuSelection);
00289             tv_valeurCourante.setText(valueOf(ruche.
getEnseleillement()) + "Watt/m²");
00290             tv_donneesBas.setText(valueOf(ruche.
getEnseleillement_Basse()) + "Watt/m²");
00291             tv_donneesHaut.setText(valueOf(ruche.
getEnseleillement_Haute()) + "Watt/m²");
00292             tv_donneesMoyenne.setText(String.format("%.2f Watt/m²",
ruche.getEnseleillement_Moyenne()));
00293             enseleillement_afficher = true;;
00294             graphiqueTemperatures.addSeries(
mSeriesEnseleillement);
00295         }
00296     }
00297 });
00298
00299 }

```

### 9.15.1.6 onCreate()

```

void fr.campus.laurainc.honeybee.GraphActivity.onCreate (
    Bundle savedInstanceState ) [protected]

```

Références [fr.campus.laurainc.honeybee.GraphActivity.handler](#), et [fr.campus.laurainc.honeybee.Ruche.recupererListeRuches\(\)](#).

```

00078         {
00079             super.onCreate(savedInstanceState);
00080             setContentView(R.layout.activity_graph);
00081
00082             choixRuche = findViewById(R.id.choixRuche);
00083             ruche = new Ruche(handler);
00084             ruche.recupererListeRuches();
00085
00086             graphiqueTemperatures = (GraphView) findViewById(R.id.graph);
00087
00088             //graphiqueTemperatures.setTitleTextSize(40);
00089             //graphiqueTemperatures.setBackgroundResource(R.color.BleuFond);
00090
00091             graphiqueTemperatures.getViewPort().setYAxisBoundsManual(true);
00092             graphiqueTemperatures.getViewPort().setMinY(0);
00093             graphiqueTemperatures.getViewPort().setMaxY(50);
00094
00095             graphiqueTemperatures.getViewPort().setMinX(0);
00096             graphiqueTemperatures.getViewPort().setMaxX(23);
00097             graphiqueTemperatures.getViewPort().setXAxisBoundsManual(true);
00098             graphiqueTemperatures.getGridLabelRenderer().setLabelFormatter(new
TimeAsXAxisLabelFormatter("H'h'")); // 0h 1h 2h ...
00099             graphiqueTemperatures.getGridLabelRenderer().setNumHorizontalLabels(12); // nb
d'heures affichées sur l'axe X
00100             graphiqueTemperatures.getGridLabelRenderer().setGridColor(Color.BLUE);
00101             graphiqueTemperatures.getGridLabelRenderer().setVerticalLabelsColor(Color.BLUE
);
00102             graphiqueTemperatures.getGridLabelRenderer().setHorizontalLabelsColor(Color.
BLUE);
00103             graphiqueTemperatures.getGridLabelRenderer().setGridStyle(GridLabelRenderer.
GridStyle.HORIZONTAL);
00104
00105             // Légende

```



```
00106         graphiqueTemperatures.getLegendRenderer().setVisible(true);
00107         //graphiqueTemperatures.getLegendRenderer().setAlign(LegendRenderer.LegendAlign.TOP);
00108         graphiqueTemperatures.getLegendRenderer().setFixedPosition(1,48);
00109         graphiqueTemperatures.getLegendRenderer().setBackgroundColor(0);
00110
00111
00112         /*GraphView graph = (GraphView) findViewById(R.id.graph);
00113
00114         LineGraphSeries<DataPoint> series = new LineGraphSeries<>(new DataPoint[] {
00115             new DataPoint(0, 1),
00116             new DataPoint(1, 5),
00117             new DataPoint(2, 3),
00118             new DataPoint(3, 2),
00119             new DataPoint(4, 6)
00120         });
00121         graph.addSeries(series);*/
00122     }
```

## 9.15.2 Documentation des données membres

### 9.15.2.1 choixRuche

Spinner fr.campus.laurainc.honeybee.GraphActivity.choixRuche [private]

### 9.15.2.2 clientMQTT

ClientMQTT fr.campus.laurainc.honeybee.GraphActivity.clientMQTT [private]

### 9.15.2.3 ensoleillement\_afficher

boolean fr.campus.laurainc.honeybee.GraphActivity.ensoleillement\_afficher [private]

### 9.15.2.4 graphiqueTemperatures

GraphView fr.campus.laurainc.honeybee.GraphActivity.graphiqueTemperatures [private]

### 9.15.2.5 handler

```
final Handler fr.campus.laurainc.honeybee.GraphActivity.handler [private]
```

#### Valeur initiale :

```
= new Handler()
{
    public void handleMessage(Message msg)
    {
        super.handleMessage(msg);
        switch (msg.what)
        {
            case HoneyBee.REQUETE_SQL_LISTE_RUCHES:
                afficherListeRuches();
            case HoneyBee.REQUETE_SQL_MESURES_RUCHES:
                Log.d(TAG, "handleMessage -> REQUETE SQL MESURES RUCHE");
                initialisationElementsIHM();
                initialiserCallBacks();
                break;
            case HoneyBee.REQUETE_SQL_RUCHE:
                ruche.recupererMesuresJournalieresRuche(
ruche.getIdRuche());
                ruche.recupererMoyennes(ruche.
getIdRuche());
        }
    }
}
```

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.afficherListeRuches\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.onCreate\(\)](#).

### 9.15.2.6 humidite\_ext\_afficher

```
boolean fr.campus.laurainc.honeybee.GraphActivity.humidite_ext_afficher [private]
```

### 9.15.2.7 humidite\_int\_afficher

```
boolean fr.campus.laurainc.honeybee.GraphActivity.humidite_int_afficher [private]
```

### 9.15.2.8 iv\_enseillement

```
Button fr.campus.laurainc.honeybee.GraphActivity.iv_enseillement [private]
```

### 9.15.2.9 iv\_humidite\_ext

```
Button fr.campus.laurainc.honeybee.GraphActivity.iv_humidite_ext [private]
```

### 9.15.2.10 iv\_humidite\_int

```
Button fr.campus.laurainc.honeybee.GraphActivity.iv_humidite_int [private]
```

**9.15.2.11 iv\_poids**

```
Button fr.campus.laurainc.honeybee.GraphActivity.iv_poids [private]
```

**9.15.2.12 iv\_pression**

```
Button fr.campus.laurainc.honeybee.GraphActivity.iv_pression [private]
```

**9.15.2.13 iv\_tem\_ext**

```
Button fr.campus.laurainc.honeybee.GraphActivity.iv_tem_ext [private]
```

**9.15.2.14 iv\_tem\_int**

```
Button fr.campus.laurainc.honeybee.GraphActivity.iv_tem_int [private]
```

**9.15.2.15 mesRuches**

```
ArrayList<String> fr.campus.laurainc.honeybee.GraphActivity.mesRuches [private]
```

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.afficherListeRuches\(\)](#).

**9.15.2.16 mSeriesEnsoleillement**

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.GraphActivity.mSeriesEnsoleillement [private]
```

**9.15.2.17 mSeriesHumiditeExt**

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.GraphActivity.mSeriesHumiditeExt [private]
```

**9.15.2.18 mSeriesHumiditeInt**

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.GraphActivity.mSeriesHumiditeInt [private]
```

**9.15.2.19 mSeriesPoids**

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.GraphActivity.mSeriesPoids [private]
```

**9.15.2.20 mSeriesPression**

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.GraphActivity.mSeriesPression [private]
```

**9.15.2.21 mSeriesTemperaturesExt**

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.GraphActivity.mSeriesTemperaturesExt [private]
```

**9.15.2.22 mSeriesTemperaturesInt**

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.GraphActivity.mSeriesTemperaturesInt [private]
```

**9.15.2.23 poids\_afficher**

```
boolean fr.campus.laurainc.honeybee.GraphActivity.poids_afficher [private]
```

**9.15.2.24 pression\_afficher**

```
boolean fr.campus.laurainc.honeybee.GraphActivity.pression_afficher [private]
```

**9.15.2.25 ruche**

```
Ruche fr.campus.laurainc.honeybee.GraphActivity.ruche [private]
```

**9.15.2.26 TAG**

```
final String fr.campus.laurainc.honeybee.GraphActivity.TAG = "GraphActivity" [static], [private]
```

**9.15.2.27 temp\_ext\_afficher**

```
boolean fr.campus.laurainc.honeybee.GraphActivity.temp_ext_afficher [private]
```

**9.15.2.28 temp\_int\_afficher**

```
boolean fr.campus.laurainc.honeybee.GraphActivity.temp_int_afficher [private]
```

**9.15.2.29 temps\_reel\_afficher**

```
boolean fr.campus.laurainc.honeybee.GraphActivity.temps_reel_afficher [private]
```

**9.15.2.30 tv\_donneesBas**

```
TextView fr.campus.laurainc.honeybee.GraphActivity.tv_donneesBas [private]
```

**9.15.2.31 tv\_donneesHaut**

```
TextView fr.campus.laurainc.honeybee.GraphActivity.tv_donneesHaut [private]
```

**9.15.2.32 tv\_donneesMoyenne**

```
TextView fr.campus.laurainc.honeybee.GraphActivity.tv_donneesMoyenne [private]
```

**9.15.2.33 tv\_valeurCourante**

```
TextView fr.campus.laurainc.honeybee.GraphActivity.tv_valeurCourante [private]
```

La documentation de cette classe a été générée à partir du fichier suivant :

— [GraphActivity.java](#)

**9.16 Référence de la classe fr.campus.laurainc.honeybee.homeActivity**

Graphe de collaboration de fr.campus.laurainc.honeybee.homeActivity :

fr.campus.laurainc.honeybee.home Activity
<ul style="list-style-type: none"><li>- layout_tableauDeBord</li><li>- layout_graphs</li><li>- layout_map</li><li>- layout_alertes</li><li>- layout_gestionAlertes</li><li>- layout_gestionRuches</li><li>- layout_debug</li><li>- img_disconnect</li></ul>
# onCreate()

## Fonctions membres protégées

— void [onCreate](#) (Bundle savedInstanceState)

## Attributs privés

— LinearLayout [layout\\_tableauDeBord](#)  
 — LinearLayout [layout\\_graphs](#)  
 — LinearLayout [layout\\_map](#)  
 — LinearLayout [layout\\_alertes](#)  
 — LinearLayout [layout\\_gestionAlertes](#)  
 — LinearLayout [layout\\_gestionRuches](#)  
 — LinearLayout [layout\\_debug](#)  
 — ImageView [img\\_disconnect](#)

## 9.16.1 Documentation des fonctions membres

## 9.16.1.1 onCreate()

```
void fr.campus.laurainc.honeybee.homeActivity.onCreate (
    Bundle savedInstanceState ) [protected]

00022                                     {
00023     super.onCreate(savedInstanceState);
00024     setContentView(R.layout.activity_home);
00025
00026     layout_tableauDeBord = findViewById(R.id.layout_tableauDeBord);
00027     layout_graphs = findViewById(R.id.layout_graphs);
00028     layout_alertes = findViewById(R.id.layout_alertes);
00029     layout_gestionAlertes = findViewById(R.id.layout_gestionAlertes);
00030     layout_gestionRuches = findViewById(R.id.layout_gestionRuche);
00031     layout_debug = findViewById(R.id.layout_debug);
00032     img_disconnect = findViewById(R.id.img_disconnect);
00033
00034
00035     final Intent tableauDeBord = new Intent(homeActivity.this, DashboardActivity.class);
00036     layout_tableauDeBord.setOnClickListener(new View.OnClickListener() {
00037         @Override
00038         public void onClick(View v)
00039         {
00040             startActivity(tableauDeBord);
00041         }
00042     });
00043
00044     final Intent graphique = new Intent(homeActivity.this, GraphActivity.class);
00045     layout_graphs.setOnClickListener(new View.OnClickListener() {
00046         @Override
00047         public void onClick(View v)
00048         {
00049             startActivity(graphique);
00050         }
00051     });
00052
00053
00054     final Intent gestionRuche = new Intent(homeActivity.this, NouvelleRucheActivity.class);
00055     layout_gestionRuches.setOnClickListener(new View.OnClickListener() {
00056         @Override
00057         public void onClick(View v)
00058         {
00059             startActivity(gestionRuche);
00060         }
00061     });
00062
00063     final Intent disconnect = new Intent(homeActivity.this, MainActivity.class);
00064     img_disconnect.setOnClickListener(new View.OnClickListener() {
00065         @Override
00066         public void onClick(View v)
00067         {
00068             startActivity(disconnect);
00069         }
00070     });
00071
00072     final Intent alertes = new Intent(homeActivity.this, alertesActivity.class);
00073     layout_alertes.setOnClickListener(new View.OnClickListener() {
00074         @Override
00075         public void onClick(View v)
00076         {
00077             startActivity(alertes);
00078         }
00079     });
00080 }
```

## 9.16.2 Documentation des données membres

### 9.16.2.1 img\_disconnect

ImageView fr.campus.laurainc.honeybee.homeActivity.img\_disconnect [private]

### 9.16.2.2 layout\_alertes

LinearLayout fr.campus.laurainc.honeybee.homeActivity.layout\_alertes [private]

### 9.16.2.3 layout\_debug

LinearLayout fr.campus.laurainc.honeybee.homeActivity.layout\_debug [private]

### 9.16.2.4 layout\_gestionAlertes

LinearLayout fr.campus.laurainc.honeybee.homeActivity.layout\_gestionAlertes [private]

### 9.16.2.5 layout\_gestionRuches

LinearLayout fr.campus.laurainc.honeybee.homeActivity.layout\_gestionRuches [private]

### 9.16.2.6 layout\_graphs

LinearLayout fr.campus.laurainc.honeybee.homeActivity.layout\_graphs [private]

### 9.16.2.7 layout\_map

LinearLayout fr.campus.laurainc.honeybee.homeActivity.layout\_map [private]

### 9.16.2.8 layout\_tableauDeBord

LinearLayout fr.campus.laurainc.honeybee.homeActivity.layout\_tableauDeBord [private]

La documentation de cette classe a été générée à partir du fichier suivant :

— [homeActivity.java](#)

## 9.17 Référence de la classe fr.campus.laurainc.honeybee.HoneyBee

Paramètres globaux de l'application.

Graphe de collaboration de fr.campus.laurainc.honeybee.HoneyBee :

fr.campus.laurainc.honeybee. HoneyBee
+ ID_Intent_ParametresHoneyBee + BDD + REQUETE_SQL_ERREUR + REQUETE_SQL_OK + REQUETE_SQL_LISTE_RUCHES + REQUETE_SQL_RUCHE + REQUETE_SQL_IDTTN + REQUETE_SQL_MESURES_RUCHES + REQUETE_SQL_ALERTES

### Attributs publics statiques

```

— static final int ID_Intent_ParametresHoneyBee = 1
    //ID de l'Intent ParametresHoneyBeeActivity
— static final boolean BDD = true
    //si vrai l'application peut utiliser la base de données MySQL (utile en debug)
— static final int REQUETE_SQL_ERREUR = -1
— static final int REQUETE_SQL_OK = 0
— static final int REQUETE_SQL_LISTE_RUCHES = 1
— static final int REQUETE_SQL_RUCHE = 2
— static final int REQUETE_SQL_IDTTN = 3
— static final int REQUETE_SQL_MESURES_RUCHES = 4
— static final int REQUETE_SQL_ALERTES = 5

```

### 9.17.1 Description détaillée

#### Auteur

Thierry Vaira

### 9.17.2 Documentation des données membres



### 9.17.2.1 BDD

```
final boolean fr.campus.laurainc.honeybee.HoneyBee.BDD = true [static]
```

Référencé par [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.deconnecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.executerRequete\(\)](#), [fr.campus.laurainc.honeybee.NouvelleRucheActivity.onCreate\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.reconnecter\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recuperer\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererChoixChAppID\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererHistoriqueAlertes\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererIdTTN\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererListeRuches\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererMesuresJournalieresRuche\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererMoyennes\(\)](#), [fr.campus.laurainc.honeybee.Ruche.Ruche\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.supprimerRuche\(\)](#).

### 9.17.2.2 ID\_Intent\_ParametresHoneyBee

```
final int fr.campus.laurainc.honeybee.HoneyBee.ID_Intent_ParametresHoneyBee = 1 [static]
```

Référencé par [fr.campus.laurainc.honeybee.MainActivity.onActivityResult\(\)](#).

### 9.17.2.3 REQUETE\_SQL\_ALERTES

```
final int fr.campus.laurainc.honeybee.HoneyBee.REQUETE_SQL_ALERTES = 5 [static]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.recupererHistoriqueAlertes\(\)](#).

### 9.17.2.4 REQUETE\_SQL\_ERREUR

```
final int fr.campus.laurainc.honeybee.HoneyBee.REQUETE_SQL_ERREUR = -1 [static]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.recuperer\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererChoixChAppID\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererHistoriqueAlertes\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererIdTTN\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererListeRuches\(\)](#), [fr.campus.laurainc.honeybee.Ruche.recupererMesuresJournalieresRuche\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.recupererMoyennes\(\)](#).

### 9.17.2.5 REQUETE\_SQL\_IDTTN

```
final int fr.campus.laurainc.honeybee.HoneyBee.REQUETE_SQL_IDTTN = 3 [static]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.recupererIdTTN\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.recupererMoyennes\(\)](#).

### 9.17.2.6 REQUETE\_SQL\_LISTE\_RUCHES

```
final int fr.campus.laurainc.honeybee.HoneyBee.REQUETE_SQL_LISTE_RUCHES = 1 [static]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.recupererChoixChAppID\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.recupererListeRuches\(\)](#).

### 9.17.2.7 REQUETE\_SQL\_MESURES\_RUCHES

```
final int fr.campus.laurainc.honeybee.HoneyBee.REQUETE_SQL_MESURES_RUCHES = 4 [static]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.recupererMesuresJournalieresRuche\(\)](#).

### 9.17.2.8 REQUETE\_SQL\_OK

```
final int fr.campus.laurainc.honeybee.HoneyBee.REQUETE_SQL_OK = 0 [static]
```

### 9.17.2.9 REQUETE\_SQL\_RUCHE

```
final int fr.campus.laurainc.honeybee.HoneyBee.REQUETE_SQL_RUCHE = 2 [static]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.recuperer\(\)](#).

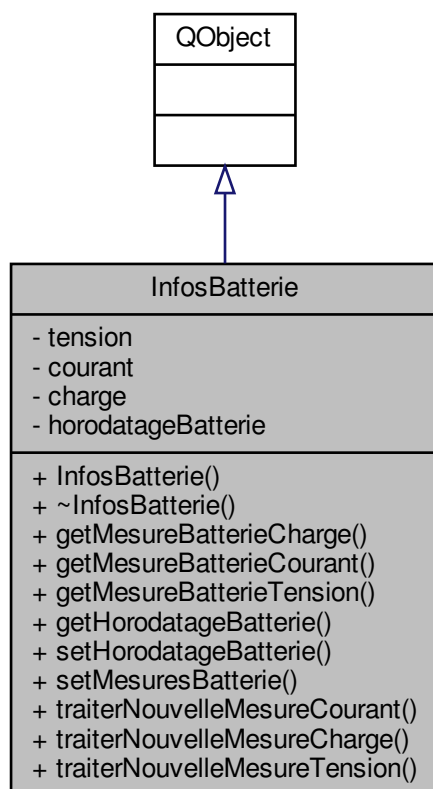
La documentation de cette classe a été générée à partir du fichier suivant :  
 — [HoneyBee.java](#)

## 9.18 Référence de la classe InfosBatterie

Déclaration de la classe [InfosBatterie](#).

```
#include <infosBatterie.h>
```

Graphe de collaboration de InfosBatterie :



## Connecteurs publics

- void `traiterNouvelleMesureCourant` (QString nouveauCourant, QString horodatage)
- void `traiterNouvelleMesureCharge` (QString nouvelleCharge, QString horodatage)
- void `traiterNouvelleMesureTension` (QString nouvelleTension, QString horodatage)

## Signaux

- void `tensionEnvoye` (const double tension, QString)
- void `courantEnvoye` (const double courant, QString)
- void `chargeEnvoye` (const double charge, QString)

## Fonctions membres publiques

- `InfosBatterie` (QObject \*parent)
- `~InfosBatterie` ()
- double `getMesureBatterieCharge` ()
- double `getMesureBatterieCourant` ()
- double `getMesureBatterieTension` ()
- QString `getHorodatageBatterie` ()
- void `setHorodatageBatterie` (QString horodatageBatterie)
- void `setMesuresBatterie` (double tension, double courant, double charge)

## Attributs privés

- double `tension`
- double `courant`
- double `charge`
- QString `horodatageBatterie`

## 9.18.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 9.18.2 Documentation des constructeurs et destructeur

## 9.18.2.1 InfosBatterie()

```
InfosBatterie::InfosBatterie (
    QObject * parent )

00016                                     : QObject(parent), tension(0),
    courant(0), charge(0), horodatageBatterie("")
00017 {
00018 }
```

### 9.18.2.2 ~InfosBatterie()

InfosBatterie::~~InfosBatterie ( )

```
00021 {  
00022  
00023 }
```

### 9.18.3 Documentation des fonctions membres

#### 9.18.3.1 chargeEnvoye

```
void InfosBatterie::chargeEnvoye (  
    const double charge,  
    QString ) [signal]
```

Référencé par [traiterNouvelleMesureCharge\(\)](#).

#### 9.18.3.2 courantEnvoye

```
void InfosBatterie::courantEnvoye (  
    const double courant,  
    QString ) [signal]
```

#### 9.18.3.3 getHorodatageBatterie()

QString InfosBatterie::getHorodatageBatterie ( )

Références [horodatageBatterie](#).

Référencé par [Ruche : :insererDonneesPortBatterie\(\)](#).

```
00041 {  
00042     return horodatageBatterie;  
00043 }
```

#### 9.18.3.4 getMesureBatterieCharge()

double InfosBatterie::getMesureBatterieCharge ( )

Références [charge](#).

Référencé par [Alertes : :alertesBatterie\(\)](#).

```
00026 {  
00027     return charge;  
00028 }
```

### 9.18.3.5 getMesureBatterieCourant()

```
double InfosBatterie::getMesureBatterieCourant ( )
```

Références [courant](#).

```
00031 {  
00032     return courant;  
00033 }
```

### 9.18.3.6 getMesureBatterieTension()

```
double InfosBatterie::getMesureBatterieTension ( )
```

Références [tension](#).

```
00036 {  
00037     return tension;  
00038 }
```

### 9.18.3.7 setHorodatageBatterie()

```
void InfosBatterie::setHorodatageBatterie (   
        QString horodatageBatterie )
```

Références [horodatageBatterie](#).

```
00046 {  
00047     this->horodatageBatterie = horodatageBatterie;  
00048 }
```

### 9.18.3.8 setMesuresBatterie()

```
void InfosBatterie::setMesuresBatterie (   
        double tension,  
        double courant,  
        double charge )
```

Références [charge](#), [courant](#), et [tension](#).

```
00051 {  
00052     this->tension = tension;  
00053     this->courant = courant;  
00054     this->charge = charge;  
00055 }
```

### 9.18.3.9 tensionEnvoye

```
void InfosBatterie::tensionEnvoye (
    const double tension,
    QString ) [signal]
```

Référencé par [traiterNouvelleMesureCourant\(\)](#), et [traiterNouvelleMesureTension\(\)](#).

### 9.18.3.10 traiterNouvelleMesureCharge

```
void InfosBatterie::traiterNouvelleMesureCharge (
    QString nouvelleCharge,
    QString horodatage ) [slot]
```

Références [charge](#), [chargeEnvoye\(\)](#), et [horodatageBatterie](#).

```
00067 {
00068     charge = nouvelleCharge.toDouble();
00069     horodatageBatterie = horodatage;
00070     emit chargeEnvoye(charge,horodatage);
00071     qDebug() << Q_FUNC_INFO << "charge = " << charge;
00072 }
```

### 9.18.3.11 traiterNouvelleMesureCourant

```
void InfosBatterie::traiterNouvelleMesureCourant (
    QString nouveauCourant,
    QString horodatage ) [slot]
```

Références [courant](#), [horodatageBatterie](#), et [tensionEnvoye\(\)](#).

```
00058 {
00059     courant = nouveauCourant.toDouble();
00060     horodatageBatterie = horodatage;
00061     emit tensionEnvoye(courant,horodatage);
00062     qDebug() << Q_FUNC_INFO << "courant = " << courant;
00063
00064 }
```

### 9.18.3.12 traiterNouvelleMesureTension

```
void InfosBatterie::traiterNouvelleMesureTension (
    QString nouvelleTension,
    QString horodatage ) [slot]
```

Références [horodatageBatterie](#), [tension](#), et [tensionEnvoye\(\)](#).

```
00075 {
00076     tension = nouvelleTension.toDouble();
00077     horodatageBatterie = horodatage;
00078     emit tensionEnvoye(tension,horodatage);
00079     qDebug() << Q_FUNC_INFO << "tension = " << tension;
00080 }
```

## 9.18.4 Documentation des données membres

## 9.18.4.1 charge

```
double InfosBatterie::charge [private]
```

Référencé par [getMesureBatterieCharge\(\)](#), [setMesuresBatterie\(\)](#), et [traiterNouvelleMesureCharge\(\)](#).

## 9.18.4.2 courant

```
double InfosBatterie::courant [private]
```

Référencé par [getMesureBatterieCourant\(\)](#), [setMesuresBatterie\(\)](#), et [traiterNouvelleMesureCourant\(\)](#).

## 9.18.4.3 horodatageBatterie

```
QString InfosBatterie::horodatageBatterie [private]
```

Référencé par [getHorodatageBatterie\(\)](#), [setHorodatageBatterie\(\)](#), [traiterNouvelleMesureCharge\(\)](#), [traiterNouvelleMesureCourant\(\)](#), et [traiterNouvelleMesureTension\(\)](#).

## 9.18.4.4 tension

```
double InfosBatterie::tension [private]
```

Référencé par [getMesureBatterieTension\(\)](#), [setMesuresBatterie\(\)](#), et [traiterNouvelleMesureTension\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

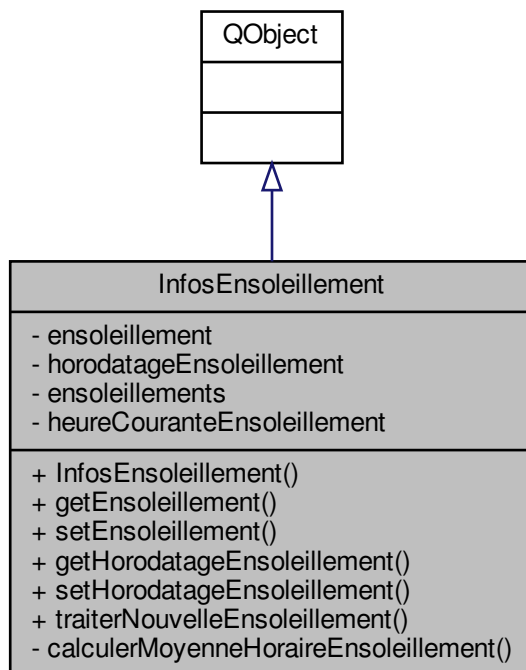
- [infosBatterie.h](#)
- [infosBatterie.cpp](#)

## 9.19 Référence de la classe InfosEnsoleillement

La classe [InfosEnsoleillement](#).

```
#include <infosEnsoleillement.h>
```

Graphique de collaboration de InfosEnsoleillement :



### Connecteurs publics

- void [traiterNouvelleEnsoleillement](#) (QString ensoleillementString, QString [horodatageEnsoleillement](#))  
*slot qui traite l'ensoleillement*

### Signaux

- void [ensoleillementEnvoye](#) (double [ensoleillement](#), QString horodatage)
- void [traitementEnsoleillementEnvoye](#) (double ensoleillementMoyen, double ensoleillementMinimum, double ensoleillementMaximum, int heure)

### Fonctions membres publiques

- [InfosEnsoleillement](#) (QObject \*parent)  
*Constructeur de la classe [InfosEnsoleillement](#).*
- double [getEnsoleillement](#) ()  
*getter de l'attribut ensoleillement*
- void [setEnsoleillement](#) (double [ensoleillement](#))  
*setter de l'attribut ensoleillement*
- QString [getHorodatageEnsoleillement](#) () const  
*getter de l'attribut horodatageEnsoleillement*
- void [setHorodatageEnsoleillement](#) (const QString [horodatageEnsoleillement](#))  
*setter de l'attribut horodatageEnsoleillement*



## Fonctions membres privées

— void `calculerMoyenneHoraireEnsoleillement` ()

## Attributs privés

— double `ensoleillement`  
*valeur courante de l'ensoleillement en w/m²*  
 — QString `horodatageEnsoleillement`  
*horodatage de la l'ensoleillement*  
 — QVector< double > `ensoleillements`  
 — int `heureCouranteEnsoleillement`

## 9.19.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 9.19.2 Documentation des constructeurs et destructeur

## 9.19.2.1 InfosEnsoleillement()

```
InfosEnsoleillement::InfosEnsoleillement (
    QObject * parent )
```

Définition des attributs `ensoleillement` à 0 et l'attribut `horodatageEnsoleillement` à ""

```
00026                                     :QObject (parent),
    ensoleillement(0.), horodatageEnsoleillement(""),
    heureCouranteEnsoleillement(-1)
00027 {
00028 }
```

## 9.19.3 Documentation des fonctions membres

### 9.19.3.1 calculerMoyenneHoraireEnsoleillement()

```
void InfosEnsoleillement::calculerMoyenneHoraireEnsoleillement ( ) [private]
```

Références [ensoleillements](#), [heureCouranteEnsoleillement](#), et [traitementEnsoleillementEnvoye](#)().

Référencé par [traiterNouvelleEnsoleillement](#)().

```
00106 {
00107     double sommeEnsoleillement= 0;
00108     double ensoleillementMoyen = 0;
00109     double ensoleillementMinimum = 999;
00110     double ensoleillementMaximum = -999;
00111
00112     // au moins 2 mesures
00113     if(ensoleillements.size() >= 2)
00114     {
00115         ensoleillementMinimum = ensoleillements[0];
00116         ensoleillementMaximum = ensoleillements[0];
00117         for (int i = 0; i < ensoleillements.size(); i++)
00118         {
00119             sommeEnsoleillement += ensoleillements[i];
00120
00121             if(ensoleillementMinimum > ensoleillements[i])
00122             {
00123                 ensoleillementMinimum = ensoleillements[i];
00124             }
00125
00126             if(ensoleillementMaximum < ensoleillements[i])
00127             {
00128                 ensoleillementMaximum = ensoleillements[i];
00129             }
00130         }
00131     }
00132     qDebug() << Q_FUNC_INFO << ensoleillements;
00133     ensoleillementMoyen = sommeEnsoleillement/ double(ensoleillements.size());
00134     emit traitementEnsoleillementEnvoye(ensoleillementMoyen,
00135         ensoleillementMinimum ,ensoleillementMaximum, heureCouranteEnsoleillement);
00136     ensoleillements.clear();
00137 }
```

### 9.19.3.2 ensoleillementEnvoye

```
void InfosEnsoleillement::ensoleillementEnvoye (
    double ensoleillement,
    QString horodatage ) [signal]
```

Référencé par [traiterNouvelleEnsoleillement](#)().

### 9.19.3.3 getEnsoleillement()

```
double InfosEnsoleillement::getEnsoleillement ( )
```

#### Renvoie

un *double* correspondant à la valeur de l'attribut ensoleillement

Références [ensoleillement](#).

Référencé par [Alertes : :alertesEnsoleillement](#)().

```
00037 {
00038     return ensoleillement;
00039 }
```

#### 9.19.3.4 getHorodatageEnseignement()

```
QString InfosEnseignement::getHorodatageEnseignement ( ) const
```

##### Renvoie

un QString correspondant a la valeur de l'attribut horodatageEnseignement

Références [horodatageEnseignement](#).

Référencé par [Ruche : :insérerDonnéesPortEnseignement\(\)](#).

```
00059 {  
00060     return horodatageEnseignement;  
00061 }
```

#### 9.19.3.5 setEnseignement()

```
void InfosEnseignement::setEnseignement (  
    double enseignement )
```

##### Paramètres

<i>enseignement</i>	
---------------------	--

Références [enseignement](#).

```
00048 {  
00049     this->enseignement = enseignement;  
00050 }
```

#### 9.19.3.6 setHorodatageEnseignement()

```
void InfosEnseignement::setHorodatageEnseignement (  
    const QString horodatageEnseignement )
```

##### Paramètres

<i>horodatageEnseignement</i>	
-------------------------------	--

Références [horodatageEnseignement](#).

```
00070 {  
00071     this->horodatageEnseignement =  
        horodatageEnseignement;  
00072 }
```

### 9.19.3.7 traitementEnsoleillementEnvoye

```
void InfosEnsoleillement::traitementEnsoleillementEnvoye (
    double ensoleillementMoyen,
    double ensoleillementMinimum,
    double ensoleillementMaximum,
    int heure ) [signal]
```

Référencé par [calculerMoyenneHoraireEnsoleillement\(\)](#).

### 9.19.3.8 traiterNouvelleEnsoleillement

```
void InfosEnsoleillement::traiterNouvelleEnsoleillement (
    QString ensoleillementString,
    QString horodatageEnsoleillement ) [slot]
```

#### Paramètres

<i>ensoleillementString</i>	qui corespond a l'ensoleillement envoyé par la <a href="#">Ruche</a>
<i>horodatageEnsoleillement</i>	correspond a l'horodatage de la mesure de l'ensoleillement

Références [calculerMoyenneHoraireEnsoleillement\(\)](#), [ensoleillement](#), [ensoleillementEnvoye\(\)](#), [ensoleillements](#), [heureCourante](#)↔  
[Ensoleillement](#), et [horodatageEnsoleillement](#).

```
00082 {
00083     ensoleillement = ensoleillementString.toDouble();
00084     this->horodatageEnsoleillement =
00085     horodatageEnsoleillement;
00086     emit ensoleillementEnvoye(ensoleillement,
00087     horodatageEnsoleillement);
00088     QDateTime dateTimeHorodatage = QDateTime::fromString(
00089     horodatageEnsoleillement, "dd/MM/yyyy HH:mm:ss");
00090     qDebug() << Q_FUNC_INFO << "heureCouranteEnsoleillement" <<
00091     heureCouranteEnsoleillement << dateTimeHorodatage.time().hour();
00092     if(heureCouranteEnsoleillement == -1)
00093     {
00094         heureCouranteEnsoleillement = dateTimeHorodatage.time().hour();
00095     }
00096     if(heureCouranteEnsoleillement == dateTimeHorodatage.time().hour())
00097     {
00098         ensoleillements.append(ensoleillement);
00099     }
00100     else if((heureCouranteEnsoleillement+1)%24 == dateTimeHorodatage.time().hour
00101     ())
00102     {
00103         calculerMoyenneHoraireEnsoleillement();
00104         heureCouranteEnsoleillement = dateTimeHorodatage.time().hour();
00105         ensoleillements.append(ensoleillement);
00106     }
00107 }
```

## 9.19.4 Documentation des données membres

### 9.19.4.1 ensoleillement

```
double InfosEnsoleillement::ensoleillement [private]
```

Référencé par [getEnsoleillement\(\)](#), [setEnsoleillement\(\)](#), et [traiterNouvelleEnsoleillement\(\)](#).

## 9.19.4.2 ensoleillements

```
QVector<double> InfosEnsoleillement::ensoleillements [private]
```

Référencé par [calculerMoyenneHoraireEnsoleillement\(\)](#), et [traiterNouvelleEnsoleillement\(\)](#).

## 9.19.4.3 heureCouranteEnsoleillement

```
int InfosEnsoleillement::heureCouranteEnsoleillement [private]
```

Référencé par [calculerMoyenneHoraireEnsoleillement\(\)](#), et [traiterNouvelleEnsoleillement\(\)](#).

## 9.19.4.4 horodatageEnsoleillement

```
QString InfosEnsoleillement::horodatageEnsoleillement [private]
```

Référencé par [getHorodatageEnsoleillement\(\)](#), [setHorodatageEnsoleillement\(\)](#), et [traiterNouvelleEnsoleillement\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

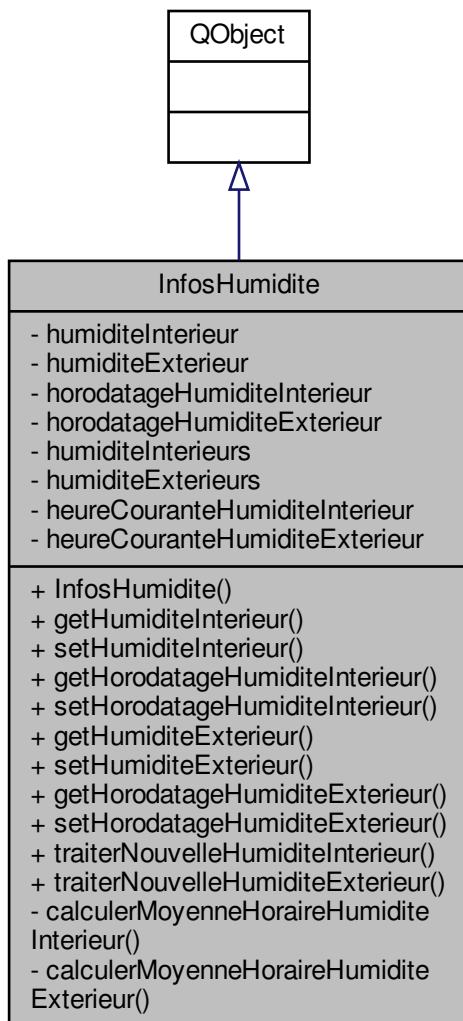
- [infosEnsoleillement.h](#)
- [infosEnsoleillement.cpp](#)

## 9.20 Référence de la classe InfosHumidite

La classe [InfosHumidite](#).

```
#include <infosHumidite.h>
```

Graphe de collaboration de InfosHumidite :



#### Connecteurs publics

- void **traiterNouvelleHumiditeInterieur** (QString humiditeInterieurstring, QString **horodatageHumiditeInterieur**)  
*slot qui traite l'humidite Interieur*
- void **traiterNouvelleHumiditeExterieur** (QString humiditeExterieurstring, QString **horodatageHumiditeExterieur**)  
*slot qui traite l'humidite Exterieur*

#### Signaux

- void **humiditeInterieurEnvoye** (const double **humiditeInterieur**, QString horodatage)  
*signal vers la classe Ruche*
- void **humiditeExterieurEnvoye** (const double **humiditeExterieur**, QString horodatage)
- void **traitementHumiditeInterieurEnvoye** (const double temperatureInterieurMoyenne, const double temperatureInterieur←  
Minimum, const double temperatureInterieurMaximum, int heure)
- void **traitementHumiditeExterieurEnvoye** (const double temperatureExterieurMoyenne, const double temperatureExterieur←  
Minimum, const double temperatureExterieurMaximum, int heure)

## Fonctions membres publiques

- [InfosHumidite](#) ([QObject](#) \*parent)  
*Constructeur de la classe [InfosHumidite](#).*
- double [getHumiditeInterieur](#) ()  
*getter de l'attribut [humiditeInterieur](#)*
- void [setHumiditeInterieur](#) (double [humiditeInterieur](#))  
*setter de l'attribut [humiditeInterieur](#)*
- [QString](#) [getHorodatageHumiditeInterieur](#) () const  
*getter de l'attribut [horodatageHumiditeInterieur](#)*
- void [setHorodatageHumiditeInterieur](#) (const [QString](#) [horodatageHumiditeInterieur](#))  
*[InfosHumidite](#) : :[setHorodatageHumiditeInterieur](#).*
- double [getHumiditeExterieur](#) ()  
*getter de l'attribut [humiditeExterieur](#)*
- void [setHumiditeExterieur](#) (double [humiditeExterieur](#))  
*setter de l'attribut [humiditeExterieur](#)*
- [QString](#) [getHorodatageHumiditeExterieur](#) () const  
*getter de l'attribut [horodatageHumiditeExterieur](#)*
- void [setHorodatageHumiditeExterieur](#) (const [QString](#) [horodatageHumiditeExterieur](#))  
*setter de l'attribut [horodatageHumiditeExterieur](#)*

## Fonctions membres privées

- void [calculerMoyenneHoraireHumiditeInterieur](#) ()
- void [calculerMoyenneHoraireHumiditeExterieur](#) ()

## Attributs privés

- double [humiditeInterieur](#)  
*humidité interieur en pourcentage*
- double [humiditeExterieur](#)  
*humidité exterieur en pourcentage*
- [QString](#) [horodatageHumiditeInterieur](#)  
*horodatage de l'humidité interieur*
- [QString](#) [horodatageHumiditeExterieur](#)  
*horodatage de l'humidité exterieur*
- [QVector](#)< double > [humiditeInterieurs](#)
- [QVector](#)< double > [humiditeExterieurs](#)
- int [heureCouranteHumiditeInterieur](#)
- int [heureCouranteHumiditeExterieur](#)

## 9.20.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 9.20.2 Documentation des constructeurs et destructeur

### 9.20.2.1 InfosHumidite()

```
InfosHumidite::InfosHumidite (
    QObject * parent )
```

Définition des attributs pressionAtmospherique à 0 et l'attribut horodatagePressionAtmospherique à ""

```
00026                                     :QObject (parent),
    humiditeInterieur(0),humiditeExterieur(0),
    horodatageHumiditeInterieur(""),
    horodatageHumiditeExterieur(""),
    heureCouranteHumiditeInterieur(-1),
    heureCouranteHumiditeExterieur(-1)
00027 {
00028 }
```

### 9.20.3 Documentation des fonctions membres

#### 9.20.3.1 calculerMoyenneHoraireHumiditeExterieur()

```
void InfosHumidite::calculerMoyenneHoraireHumiditeExterieur ( ) [private]
```

Références [heureCouranteHumiditeExterieur](#), [humiditeExterieurs](#), et [traitementHumiditeExterieurEnvoye\(\)](#).

Référencé par [traiterNouvelleHumiditeExterieur\(\)](#).

```
00215 {
00216     double sommeHumiditeExterieur = 0;
00217     double humiditeExterieurMoyenne = 0;
00218     double humiditeExterieurMinimum = 999;
00219     double humiditeExterieurMaximum = -999;
00220
00221     // au moins 2 mesures
00222     if(humiditeExterieurs.size() >= 2)
00223     {
00224         humiditeExterieurMinimum = humiditeExterieurs[0];
00225         humiditeExterieurMaximum = humiditeExterieurs[0];
00226         for (int i = 0; i < humiditeExterieurs.size(); i++)
00227         {
00228             sommeHumiditeExterieur += humiditeExterieurs[i];
00229
00230             if(humiditeExterieurMinimum > humiditeExterieurs[i])
00231             {
00232                 humiditeExterieurMinimum = humiditeExterieurs[i];
00233             }
00234
00235             if(humiditeExterieurMaximum < humiditeExterieurs[i])
00236             {
00237                 humiditeExterieurMaximum = humiditeExterieurs[i];
00238             }
00239         }
00240     }
00241     qDebug() << Q_FUNC_INFO << humiditeExterieurs;
00242     qDebug() << Q_FUNC_INFO << "humiditeExterieurMoyenne=" << humiditeExterieurMoyenne << "
    humiditeExterieurMinimum=" << humiditeExterieurMinimum << "humiditeExterieurMaximum=" << humiditeExterieurMaximum;
00243     humiditeExterieurMoyenne = sommeHumiditeExterieur / double(humiditeExterieurs.size());
00244     emit traitementHumiditeExterieurEnvoye(humiditeExterieurMoyenne,
    humiditeExterieurMinimum, humiditeExterieurMaximum, heureCouranteHumiditeExterieur)
    ;
00245     humiditeExterieurs.clear();
00246 }
```



## 9.20.3.2 calculerMoyenneHoraireHumiditeInterieur()

```
void InfosHumidite::calculerMoyenneHoraireHumiditeInterieur ( ) [private]
```

Références [heureCouranteHumiditeInterieur](#), [humiditeInterieurs](#), et [traitementHumiditeInterieurEnvoye](#)).

Référencé par [traiterNouvelleHumiditeInterieur](#)).

```
00176 {
00177     double sommeHumiditeInterieur = 0;
00178     double humiditeInterieurMoyenne = 0;
00179     double humiditeInterieurMinimum = 999;
00180     double humiditeInterieurMaximum = -999;
00181
00182     // au moins 2 mesures
00183     if(humiditeInterieurs.size() >= 2)
00184     {
00185         humiditeInterieurMinimum = humiditeInterieurs[0];
00186         humiditeInterieurMaximum = humiditeInterieurs[0];
00187         for (int i = 0; i < humiditeInterieurs.size(); i++)
00188         {
00189             sommeHumiditeInterieur += humiditeInterieurs[i];
00190
00191             if(humiditeInterieurMinimum > humiditeInterieurs[i])
00192             {
00193                 humiditeInterieurMinimum = humiditeInterieurs[i];
00194             }
00195
00196             if(humiditeInterieurMaximum < humiditeInterieurs[i])
00197             {
00198                 humiditeInterieurMaximum = humiditeInterieurs[i];
00199             }
00200         }
00201     }
00202     qDebug() << Q_FUNC_INFO << humiditeInterieurs;
00203     qDebug() << Q_FUNC_INFO << "humiditeInterieurMoyenne=" << humiditeInterieurMoyenne << "
humiditeInterieurMinimum=" << humiditeInterieurMinimum << "humiditeInterieurMaximum=" << humiditeInterieurMaximum;
00204     humiditeInterieurMoyenne = sommeHumiditeInterieur / double(humiditeInterieurs.size());
00205     emit traitementHumiditeInterieurEnvoye(humiditeInterieurMoyenne,
humiditeInterieurMinimum, humiditeInterieurMaximum, heureCouranteHumiditeInterieur)
;
00206     humiditeInterieurs.clear();
00207 }
```

## 9.20.3.3 getHorodatageHumiditeExterieur()

```
QString InfosHumidite::getHorodatageHumiditeExterieur ( ) const
```

## Renvoi

un *QString* correspondant a la valeur de l'attribut horodatageHumiditeExterieur

Références [horodatageHumiditeExterieur](#).

```
00098 {
00099     return horodatageHumiditeExterieur;
00100 }
```

## 9.20.3.4 getHorodatageHumiditeInterieur()

```
QString InfosHumidite::getHorodatageHumiditeInterieur ( ) const
```

## Renvoi

*QString* horodatageHumiditeInterieur correspondant a l'attribut horodatageHumiditeInterieur

Références [horodatageHumiditeInterieur](#).

```
00057 {
00058     return horodatageHumiditeInterieur;
00059 }
```

### 9.20.3.5 getHumiditeExterieur()

```
double InfosHumidite::getHumiditeExterieur ( )
```

#### Renvoie

*double* la valeur de l'humiditeExterieur

Références [humiditeExterieur](#).

Référencé par [Alertes : :alertesHumiditeExterieur\(\)](#).

```
00077 {  
00078     return humiditeExterieur;  
00079 }
```

### 9.20.3.6 getHumiditeInterieur()

```
double InfosHumidite::getHumiditeInterieur ( )
```

#### Renvoie

*double* correspondant a la valeur de l'attribut humiditeInterieur

Références [humiditeInterieur](#).

Référencé par [Alertes : :alertesHumiditeInterieur\(\)](#), et [Ruche : :insererDonneesPortMesureRuche\(\)](#).

```
00037 {  
00038     return humiditeInterieur;  
00039 }
```

### 9.20.3.7 humiditeExterieurEnvoye

```
void InfosHumidite::humiditeExterieurEnvoye (  
    const double humiditeExterieur,  
    QString horodatage ) [signal]
```

Référencé par [traiterNouvelleHumiditeExterieur\(\)](#).

### 9.20.3.8 humiditeInterieurEnvoye

```
void InfosHumidite::humiditeInterieurEnvoye (  
    const double humiditeInterieur,  
    QString horodatage ) [signal]
```

Référencé par [traiterNouvelleHumiditeInterieur\(\)](#).

### 9.20.3.9 setHorodatageHumiditeExterieur()

```
void InfosHumidite::setHorodatageHumiditeExterieur (  
    const QString horodatageHumiditeExterieur )
```

## Paramètres

<i>horodatageHumiditeExterieur</i>	
------------------------------------	--

Références [horodatageHumiditeExterieur](#).

```
00109 {
00110     this->horodatageHumiditeExterieur =
        horodatageHumiditeExterieur;
00111 }
```

## 9.20.3.10 setHorodatageHumiditeInterieur()

```
void InfosHumidite::setHorodatageHumiditeInterieur (
    const QString horodatageHumiditeInterieur )
```

## Paramètres

<i>horodatageHumiditeInterieur</i>	correspondant a l'horodatage humiddite interieur
------------------------------------	--

Références [horodatageHumiditeInterieur](#).

```
00067 {
00068     this->horodatageHumiditeInterieur =
        horodatageHumiditeInterieur;
00069 }
```

## 9.20.3.11 setHumiditeExterieur()

```
void InfosHumidite::setHumiditeExterieur (
    double humiditeExterieur )
```

## Paramètres

<i>humiditeExterieur</i>	corespondant a l'humidité l'exterieur
--------------------------	---------------------------------------

Références [humiditeExterieur](#).

```
00087 {
00088     this->humiditeExterieur = humiditeExterieur;
00089 }
```

## 9.20.3.12 setHumiditeInterieur()

```
void InfosHumidite::setHumiditeInterieur (
    double humiditeInterieur )
```

## Paramètres

<i>humiditeInterieur</i>	correspondant a l'humidite interieur
--------------------------	--------------------------------------

Références [humiditeInterieur](#).

```
00047 {
00048     this->humiditeInterieur = humiditeInterieur;
00049 }
```

## 9.20.3.13 traitementHumiditeExterieurEnvoye

```
void InfosHumidite::traitementHumiditeExterieurEnvoye (
    const double temperatureExterieurMoyenne,
    const double temperatureExterieurMinimum,
    const double temperatureExterieurMaximum,
    int heure ) [signal]
```

Référencé par [calculerMoyenneHoraireHumiditeExterieur\(\)](#).

## 9.20.3.14 traitementHumiditeInterieurEnvoye

```
void InfosHumidite::traitementHumiditeInterieurEnvoye (
    const double temperatureInterieurMoyenne,
    const double temperatureInterieurMinimum,
    const double temperatureInterieurMaximum,
    int heure ) [signal]
```

Référencé par [calculerMoyenneHoraireHumiditeInterieur\(\)](#).

## 9.20.3.15 traiterNouvelleHumiditeExterieur

```
void InfosHumidite::traiterNouvelleHumiditeExterieur (
    QString humiditeExterieurstring,
    QString horodatageHumiditeExterieur ) [slot]
```

## Paramètres

<i>humiditeExterieurstring</i>	qui corespond a l'humidite exterieur envoyé par la <a href="#">Ruche</a>
<i>horodatageHumiditeExterieur</i>	qui correspond a l'horodatage de la mesure de l'humidité exterieur

Références [calculerMoyenneHoraireHumiditeExterieur\(\)](#), [heureCouranteHumiditeExterieur](#), [humiditeExterieur](#), [humiditeExterieur↔Envoye\(\)](#), et [humiditeExterieurs](#).

```
00152 {
00153     humiditeExterieur = humiditeExterieurstring.toDouble();
00154     emit humiditeExterieurEnvoye(humiditeExterieur,
00155                                horodatageHumiditeExterieur);
00155 }
```

```

00156
00157     QDateTime dateTimeHorodatage = QDateTime::fromString(
    horodatageHumiditeExterieur, "dd/MM/yyyy HH:mm:ss");
00158     qDebug() << Q_FUNC_INFO << "heureCouranteHumiditeExterieur" <<
    heureCouranteHumiditeExterieur << dateTimeHorodatage.time().hour();
00159     if(heureCouranteHumiditeExterieur == -1)
00160     {
00161         heureCouranteHumiditeExterieur = dateTimeHorodatage.time().hour();
00162     }
00163     if(heureCouranteHumiditeExterieur == dateTimeHorodatage.time().hour())
00164     {
00165         humiditeExterieurs.append(humiditeExterieur);
00166     }
00167     else if((heureCouranteHumiditeExterieur+1)%24 == dateTimeHorodatage.time(
).hour())
00168     {
00169         calculerMoyenneHoraireHumiditeExterieur();
00170         heureCouranteHumiditeExterieur = dateTimeHorodatage.time().hour();
00171         humiditeExterieurs.append(humiditeExterieur);
00172     }
00173 }

```

### 9.20.3.16 traiterNouvelleHumiditeInterieur

```

void InfosHumidite::traiterNouvelleHumiditeInterieur (
    QString humiditeInterieurstring,
    QString horodatageHumiditeInterieur ) [slot]

```

#### Paramètres

<i>humiditeInterieurstring</i>	qui correspond a l'humidite interieur envoyé par la <a href="#">Ruche</a>
<i>horodatageHumiditeInterieur</i>	correspond a l'horodatage de la mesure de l'humidité interieur

Références [calculerMoyenneHoraireHumiditeInterieur\(\)](#), [heureCouranteHumiditeInterieur](#), [humiditeInterieur](#), [humiditeInterieur←Envoye\(\)](#), et [humiditeInterieurs](#).

```

00121 {
00122     humiditeInterieur = humiditeInterieurstring.toDouble();
00123     emit humiditeInterieurEnvoye(humiditeInterieur,
    horodatageHumiditeInterieur);
00124
00125     QDateTime dateTimeHorodatage = QDateTime::fromString(
    horodatageHumiditeInterieur, "dd/MM/yyyy HH:mm:ss");
00126     qDebug() << Q_FUNC_INFO << "heureCouranteHumiditeInterieur" <<
    heureCouranteHumiditeInterieur << dateTimeHorodatage.time().hour();
00127     if(heureCouranteHumiditeInterieur == -1)
00128     {
00129         heureCouranteHumiditeInterieur = dateTimeHorodatage.time().hour();
00130     }
00131     if(heureCouranteHumiditeInterieur == dateTimeHorodatage.time().hour())
00132     {
00133         humiditeInterieurs.append(humiditeInterieur);
00134     }
00135     else if((heureCouranteHumiditeInterieur+1)%24 == dateTimeHorodatage.time(
).hour())
00136     {
00137         calculerMoyenneHoraireHumiditeInterieur();
00138         heureCouranteHumiditeInterieur = dateTimeHorodatage.time().hour();
00139         humiditeInterieurs.append(humiditeInterieur);
00140     }
00141 }

```

### 9.20.4 Documentation des données membres

#### 9.20.4.1 heureCouranteHumiditeExterieur

```
int InfosHumidite::heureCouranteHumiditeExterieur [private]
```

Référencé par [calculerMoyenneHoraireHumiditeExterieur\(\)](#), et [traiterNouvelleHumiditeExterieur\(\)](#).

#### 9.20.4.2 heureCouranteHumiditeInterieur

```
int InfosHumidite::heureCouranteHumiditeInterieur [private]
```

Référencé par [calculerMoyenneHoraireHumiditeInterieur\(\)](#), et [traiterNouvelleHumiditeInterieur\(\)](#).

#### 9.20.4.3 horodatageHumiditeExterieur

```
QString InfosHumidite::horodatageHumiditeExterieur [private]
```

Référencé par [getHorodatageHumiditeExterieur\(\)](#), et [setHorodatageHumiditeExterieur\(\)](#).

#### 9.20.4.4 horodatageHumiditeInterieur

```
QString InfosHumidite::horodatageHumiditeInterieur [private]
```

Référencé par [getHorodatageHumiditeInterieur\(\)](#), et [setHorodatageHumiditeInterieur\(\)](#).

#### 9.20.4.5 humiditeExterieur

```
double InfosHumidite::humiditeExterieur [private]
```

Référencé par [getHumiditeExterieur\(\)](#), [setHumiditeExterieur\(\)](#), et [traiterNouvelleHumiditeExterieur\(\)](#).

#### 9.20.4.6 humiditeExterieurs

```
QVector<double> InfosHumidite::humiditeExterieurs [private]
```

Référencé par [calculerMoyenneHoraireHumiditeExterieur\(\)](#), et [traiterNouvelleHumiditeExterieur\(\)](#).

#### 9.20.4.7 humiditeInterieur

```
double InfosHumidite::humiditeInterieur [private]
```

Référencé par [getHumiditeInterieur\(\)](#), [setHumiditeInterieur\(\)](#), et [traiterNouvelleHumiditeInterieur\(\)](#).

## 9.20.4.8 humiditeInterieurs

```
QVector<double> InfosHumidite::humiditeInterieurs [private]
```

Référencé par [calculerMoyenneHoraireHumiditeInterieur\(\)](#), et [traiterNouvelleHumiditeInterieur\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

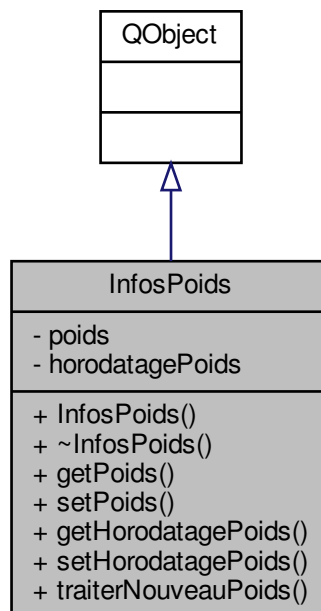
- [infosHumidite.h](#)
- [infosHumidite.cpp](#)

## 9.21 Référence de la classe InfosPoids

Déclaration de la classe [InfosPoids](#).

```
#include <infosPoids.h>
```

Graphe de collaboration de InfosPoids :



## Connecteurs publics

- void [traiterNouveauPoids](#) (QString nouveauPoids, QString [horodatagePoids](#))

## Signaux

- void [poidsEnvoye](#) (const double [poids](#), QString)

## Fonctions membres publiques

- `InfosPoids (QObject *parent)`
- `~InfosPoids ()`
- `double getPoids ()`
- `void setPoids (double poids)`
- `QString getHorodatagePoids ()`
- `void setHorodatagePoids (QString horodatagePoids)`

## Attributs privés

- `double poids`
- `QString horodatagePoids`

## 9.21.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 9.21.2 Documentation des constructeurs et destructeur

## 9.21.2.1 InfosPoids()

```
InfosPoids::InfosPoids (
    QObject * parent )
```

```
00016                                     : QObject (parent), poids (0) ,
00017                                     horodatagePoids ("")
00017 {
00018 }
```

## 9.21.2.2 ~InfosPoids()

```
InfosPoids::~InfosPoids ( )
```

```
00021 {
00022 }
```

## 9.21.3 Documentation des fonctions membres



### 9.21.3.1 getHorodatagePoids()

```
QString InfosPoids::getHorodatagePoids ( )
```

Références [horodatagePoids](#).

Référencé par [Ruche](#) : [insererDonneesPortPoids\(\)](#).

```
00043 {  
00044     return horodatagePoids;  
00045 }
```

### 9.21.3.2 getPoids()

```
double InfosPoids::getPoids ( )
```

Références [poids](#).

Référencé par [Alertes](#) : [alertesPoids\(\)](#).

```
00025 {  
00026     return poids;  
00027 }
```

### 9.21.3.3 poidsEnvoye

```
void InfosPoids::poidsEnvoye (  
    const double poids,  
    QString ) [signal]
```

Référencé par [traiterNouveauPoids\(\)](#).

### 9.21.3.4 setHorodatagePoids()

```
void InfosPoids::setHorodatagePoids (  
    QString horodatagePoids )
```

Références [horodatagePoids](#).

```
00047 {  
00048     this->horodatagePoids = horodatagePoids;  
00049 }
```

#### 9.21.3.5 setPoids()

```
void InfosPoids::setPoids (
    double poids )
```

Références [poids](#).

```
00030 {
00031     this->poids = poids;
00032 }
```

#### 9.21.3.6 traiterNouveauPoids

```
void InfosPoids::traiterNouveauPoids (
    QString nouveauPoids,
    QString horodatagePoids ) [slot]
```

Références [horodatagePoids](#), [poids](#), et [poidsEnvoye\(\)](#).

```
00035 {
00036     this->horodatagePoids = horodatagePoids;
00037     poids = nouveauPoids.toDouble() / 1000;
00038     emit poidsEnvoye(poids,horodatagePoids);
00039     qDebug() << Q_FUNC_INFO << "poids = " << poids;
00040 }
```

### 9.21.4 Documentation des données membres

#### 9.21.4.1 horodatagePoids

```
QString InfosPoids::horodatagePoids [private]
```

Référencé par [getHorodatagePoids\(\)](#), [setHorodatagePoids\(\)](#), et [traiterNouveauPoids\(\)](#).

#### 9.21.4.2 poids

```
double InfosPoids::poids [private]
```

Référencé par [getPoids\(\)](#), [setPoids\(\)](#), et [traiterNouveauPoids\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

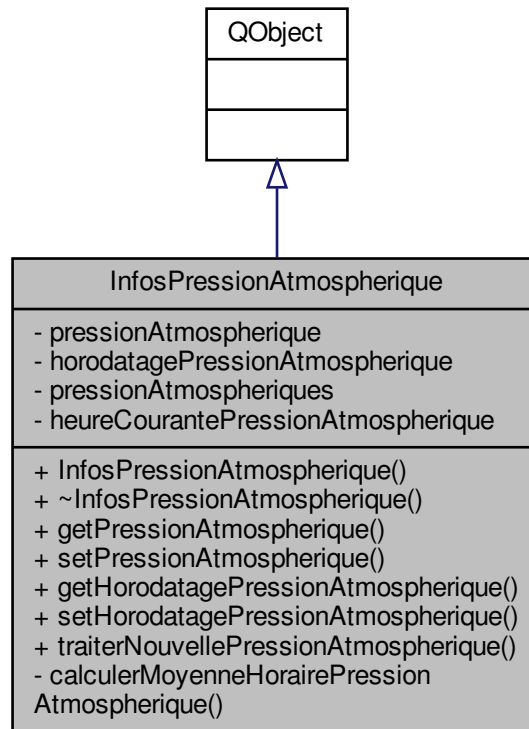
- [infosPoids.h](#)
- [infosPoids.cpp](#)

## 9.22 Référence de la classe InfosPressionAtmospherique

La classe [InfosPressionAtmospherique](#).

```
#include <infosPressionAtmospherique.h>
```

Graphe de collaboration de InfosPressionAtmospherique :



## Connecteurs publics

- void [traiterNouvellePressionAtmospherique](#) (QString pressionAtmospheriqueString, QString [horodatagePressionAtmospherique](#))  
*slot qui traite la pression atmospherique*

## Signaux

- void [pressionAtmospheriqueEnvoye](#) (double [pressionAtmospherique](#), QString horodatage)
- void [traitementPressionAtmospheriqueEnvoye](#) (const double pressionAtmospheriqueMoyenne, const double pressionAtmospheriqueMinimum, const double pressionAtmospheriqueMaximum, int heure)

## Fonctions membres publiques

- [InfosPressionAtmospherique](#) (QObject \*parent)
- [~InfosPressionAtmospherique](#) ()  
*destructeur de la classe InfosPressionAtmospherique*
- double [getPressionAtmospherique](#) ()  
*getter de l'attribut pressionAtmospherique*
- void [setPressionAtmospherique](#) (double [pressionAtmospherique](#))  
*setter de l'attribut pressionAtmospherique*
- QString [getHorodatagePressionAtmospherique](#) () const  
*getter de l'attribut horodatagePressionAtmospherique*
- void [setHorodatagePressionAtmospherique](#) (const QString [horodatagePressionAtmospherique](#))  
*setter de l'attribut horodatagePressionAtmospherique*

## Fonctions membres privées

— void `calculerMoyenneHorairePressionAtmospherique` ()

## Attributs privés

- double `pressionAtmospherique`  
*la pression atmosferique (unité ?)*
- QString `horodatagePressionAtmospherique`  
*horodatage de la pression atmosferique*
- QVector< double > `pressionAtmospheriques`
- int `heureCourantePressionAtmospherique`

## 9.22.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 9.22.2 Documentation des constructeurs et destructeur

## 9.22.2.1 InfosPressionAtmospherique()

```
InfosPressionAtmospherique::InfosPressionAtmospherique (
    QObject * parent )
```

```
00026                                     : QObject (parent),
    pressionAtmospherique(0), horodatagePressionAtmospherique
    (""),heureCourantePressionAtmospherique(-1)
00027 {
00028 }
```

## 9.22.2.2 ~InfosPressionAtmospherique()

```
InfosPressionAtmospherique::~InfosPressionAtmospherique ( )
```

```
00034 {
00035 }
```

## 9.22.3 Documentation des fonctions membres

## 9.22.3.1 calculerMoyenneHorairePressionAtmospherique()

```
void InfosPressionAtmospherique::calculerMoyenneHorairePressionAtmospherique ( ) [private]
```

Références [heureCourantePressionAtmospherique](#), [pressionAtmospheriques](#), et [traitementPressionAtmospheriqueEnvoye\(\)](#).

Référencé par [traiterNouvellePressionAtmospherique\(\)](#).

```
00105 {
00106     double sommePressionAtmospherique= 0;
00107     double pressionAtmospheriqueMoyenne = 0;
00108     double pressionAtmospheriqueMinimum = 1200;
00109     double pressionAtmospheriqueMaximum = -999;
00110
00111     // au moins 2 mesures
00112     if (pressionAtmospheriques.size() >= 2)
00113     {
00114         pressionAtmospheriqueMinimum = pressionAtmospheriques[0];
00115         pressionAtmospheriqueMaximum = pressionAtmospheriques[0];
00116         for (int i = 0; i < pressionAtmospheriques.size(); i++)
00117         {
00118             sommePressionAtmospherique += pressionAtmospheriques[i];
00119
00120             if (pressionAtmospheriqueMinimum > pressionAtmospheriques[i])
00121             {
00122                 pressionAtmospheriqueMinimum = pressionAtmospheriques[i];
00123             }
00124
00125             if (pressionAtmospheriqueMaximum < pressionAtmospheriques[i])
00126             {
00127                 pressionAtmospheriqueMaximum = pressionAtmospheriques[i];
00128             }
00129         }
00130     }
00131     qDebug() << Q_FUNC_INFO << pressionAtmospheriques;
00132     pressionAtmospheriqueMoyenne = sommePressionAtmospherique/ double(pressionAtmospheriques.size());
00133     emit traitementPressionAtmospheriqueEnvoye(
        pressionAtmospheriqueMoyenne, pressionAtmospheriqueMinimum ,pressionAtmospheriqueMaximum,
        heureCourantePressionAtmospherique);
00134     pressionAtmospheriques.clear();
00135 }
```

## 9.22.3.2 getHorodatagePressionAtmospherique()

```
QString InfosPressionAtmospherique::getHorodatagePressionAtmospherique ( ) const
```

**Renvoie**

*QString* horodatagePressionAtmospherique

Références [horodatagePressionAtmospherique](#).

```
00071 {
00072     return horodatagePressionAtmospherique;
00073 }
```

## 9.22.3.3 getPressionAtmospherique()

```
double InfosPressionAtmospherique::getPressionAtmospherique ( )
```

**Renvoie**

*double* pressionAtmospherique

Références [pressionAtmospherique](#).

Référencé par [Alertes : :alertesPressionAtmospherique\(\)](#).

```
00043 {
00044     return pressionAtmospherique;
00045 }
```

#### 9.22.3.4 pressionAtmospheriqueEnvoie

```
void InfosPressionAtmospherique::pressionAtmospheriqueEnvoie (
    double pressionAtmospherique,
    QString horodatage ) [signal]
```

Référencé par [traiterNouvellePressionAtmospherique\(\)](#).

#### 9.22.3.5 setHorodatagePressionAtmospherique()

```
void InfosPressionAtmospherique::setHorodatagePressionAtmospherique (
    const QString horodatagePressionAtmospherique )
```

##### Paramètres

<i>horodatagePressionAtmospherique</i>	correspondant a la horodatagePressionAtmospherique
--	--

Références [horodatagePressionAtmospherique](#).

```
00062 {
00063     this->horodatagePressionAtmospherique =
00064         horodatagePressionAtmospherique;
00064 }
```

#### 9.22.3.6 setPressionAtmospherique()

```
void InfosPressionAtmospherique::setPressionAtmospherique (
    double pressionAtmospherique )
```

##### Paramètres

<i>pressionAtmospherique</i>	correspondant a la pression atmospherique
------------------------------	---

Références [pressionAtmospherique](#).

```
00053 {
00054     this->pressionAtmospherique = pressionAtmospherique;
00055 }
```

#### 9.22.3.7 traitementPressionAtmospheriqueEnvoie

```
void InfosPressionAtmospherique::traitementPressionAtmospheriqueEnvoie (
    const double pressionAtmospheriqueMoyenne,
    const double pressionAtmospheriqueMinimum,
    const double pressionAtmospheriqueMaximum,
    int heure ) [signal]
```

Référencé par [calculerMoyenneHorairePressionAtmospherique\(\)](#).

## 9.22.3.8 traiterNouvellePressionAtmospherique

```
void InfosPressionAtmospherique::traiterNouvellePressionAtmospherique (
    QString pressionAtmospheriqueString,
    QString horodatagePressionAtmospherique ) [slot]
```

## Paramètres

<i>pressionAtmospheriqueString</i>	qui corespond a la pression atmospherique envoyé par la <a href="#">Ruche</a>
<i>horodatagePressionAtmospherique</i>	correspond a l'horodatage de la mesure de la pression atmospherique

Références [calculerMoyenneHorairePressionAtmospherique\(\)](#), [heureCourantePressionAtmospherique](#), [pressionAtmospherique](#), [pressionAtmospheriqueEnvoye\(\)](#), et [pressionAtmospheriques](#).

```
00082 {
00083     pressionAtmospherique = pressionAtmospheriqueString.toDouble();
00084     emit pressionAtmospheriqueEnvoye(
00085         pressionAtmospherique, horodatagePressionAtmospherique);
00086     QDateTime dateTimeHorodatage = QDateTime::fromString(
00087         horodatagePressionAtmospherique, "dd/MM/yyyy HH:mm:ss");
00088     qDebug() << Q_FUNC_INFO << "heureCourantePressionAtmospherique" <<
00089         heureCourantePressionAtmospherique << dateTimeHorodatage.time().hour();
00090     if(heureCourantePressionAtmospherique == -1)
00091     {
00092         heureCourantePressionAtmospherique = dateTimeHorodatage.time().
00093             hour();
00094     }
00095     if(heureCourantePressionAtmospherique == dateTimeHorodatage.time().
00096         hour())
00097     {
00098         pressionAtmospheriques.append(
00099             pressionAtmospherique);
00100     }
00101     else if((heureCourantePressionAtmospherique+1)%24 ==
00102         dateTimeHorodatage.time().hour())
00103     {
00104         calculerMoyenneHorairePressionAtmospherique();
00105         heureCourantePressionAtmospherique = dateTimeHorodatage.time().
00106             hour();
00107         pressionAtmospheriques.append(
00108             pressionAtmospherique);
00109     }
00110 }
```

## 9.22.4 Documentation des données membres

## 9.22.4.1 heureCourantePressionAtmospherique

```
int InfosPressionAtmospherique::heureCourantePressionAtmospherique [private]
```

Référencé par [calculerMoyenneHorairePressionAtmospherique\(\)](#), et [traiterNouvellePressionAtmospherique\(\)](#).

## 9.22.4.2 horodatagePressionAtmospherique

```
QString InfosPressionAtmospherique::horodatagePressionAtmospherique [private]
```

Référencé par [getHorodatagePressionAtmospherique\(\)](#), et [setHorodatagePressionAtmospherique\(\)](#).

#### 9.22.4.3 pressionAtmospherique

```
double InfosPressionAtmospherique::pressionAtmospherique [private]
```

Référéncé par [getPressionAtmospherique\(\)](#), [setPressionAtmospherique\(\)](#), et [traiterNouvellePressionAtmospherique\(\)](#).

#### 9.22.4.4 pressionAtmospheriques

```
QVector<double> InfosPressionAtmospherique::pressionAtmospheriques [private]
```

Référéncé par [calculerMoyenneHorairePressionAtmospherique\(\)](#), et [traiterNouvellePressionAtmospherique\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

- [infosPressionAtmospherique.h](#)
- [infosPressionAtmospherique.cpp](#)

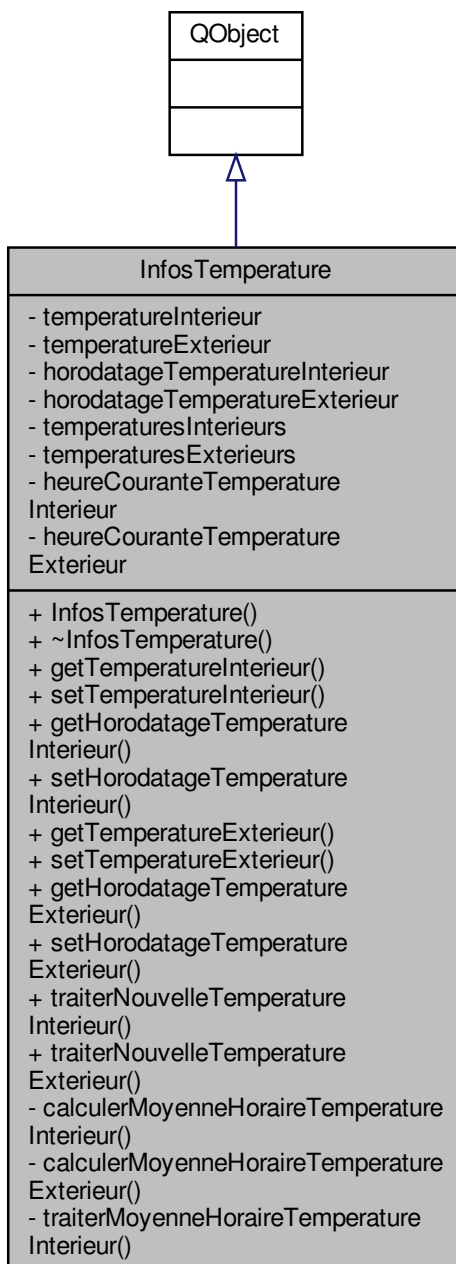
### 9.23 Référence de la classe InfosTemperature

La classe [InfosTemperature](#).

```
#include <infosTemperature.h>
```



Graphe de collaboration de InfosTemperature :



#### Connecteurs publics

- void `traiterNouvelleTemperatureInterieur` (QString `temperatureInterieur`, QString `horodatageTemperatureInterieur`)
- void `traiterNouvelleTemperatureExterieur` (QString `temperatureExterieurString`, QString `horodatageTemperatureExterieur`)  
*slot qui traite la temperature exterieur venant de la classe `Ruche`*

#### Signaux

- void `temperatureInterieurEnvoye` (const double `temperatureInterieur`, QString `horodatage`)

- *signal vers la classe Ruche*
- void `temperatureExterieurEnvoye` (const double `temperatureExterieur`, QString horodatage)
- *signal vers la classe Ruche*
- void `traitementTemperatureInterieurEnvoye` (const double `temperatureInterieurMoyenne`, const double `temperatureInterieur` ← Minimum, const double `temperatureInterieurMaximum`, int heure)
- void `traitementTemperatureExterieurEnvoye` (const double `temperatureExterieurMoyenne`, const double `temperature` ← ExterieurMinimum, const double `temperatureExterieurMaximum`, int heure)

### Fonctions membres publiques

- `InfosTemperature` (QObject \*parent)
- *Constructeur de la classe InfosPressionAtmospherique.*
- `~InfosTemperature` ()
- double `getTemperatureInterieur` ()
- *getter de l'attribut temperatureInterieur*
- void `setTemperatureInterieur` (double `temperatureInterieur`)
- QString `getHorodatageTemperatureInterieur` ()
- *getter de l'attribut horodatagetemperatureInterieur*
- void `setHorodatageTemperatureInterieur` (const QString `horodatageTemperatureInterieur`)
- double `getTemperatureExterieur` ()
- *getter de l'attribut temperatureExterieur*
- void `setTemperatureExterieur` (double `temperatureExterieur`)
- *setter de l'attribut temperatureExterieur*
- QString `getHorodatageTemperatureExterieur` ()
- void `setHorodatageTemperatureExterieur` (const QString `horodatageTemperatureExterieur`)
- *setter de l'attribut horodatageTemperatureExterieur*

### Fonctions membres privées

- void `calculerMoyenneHoraireTemperatureInterieur` ()
- *methode permettant de calculer la moyenne des temperatures interieurs sur une heure*
- void `calculerMoyenneHoraireTemperatureExterieur` ()
- *methode permettant de calculer la moyenne des temperatures exterieurs sur une heure*
- void `traiterMoyenneHoraireTemperatureInterieur` (QString `horodatageTemperatureInterieur`)
- *slot qui traite la temperature interieur venant de la classe Ruche*

### Attributs privés

- double `temperatureInterieur`
- *temperature interieur en degreé Celsius*
- double `temperatureExterieur`
- *temperature exterieuren degreé Celsius*
- QString `horodatageTemperatureInterieur`
- *horodatage de la temperature interieur*
- QString `horodatageTemperatureExterieur`
- *horodatage de la temperature exterieur*
- QVector< double > `temperaturesInterieurs`
- QVector< double > `temperaturesExterieurs`
- int `heureCouranteTemperatureInterieur`
- int `heureCouranteTemperatureExterieur`

### 9.23.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 9.23.2 Documentation des constructeurs et destructeur

## 9.23.2.1 InfosTemperature()

```
InfosTemperature::InfosTemperature (
    QObject * parent )
```

Constructeur de la classe [InfosTemperature](#).

Définition des attributs pressionAtmospherique à 0 et l'attribut horodatagePressionAtmospherique à ""

Définition des attributs temperatureInterieur, temperatureExterieur à 0 et les attributs temperatureExterieur horodatageTemperature↵  
Exterieur à ""

## Paramètres

<i>parent</i>	<a href="#">QObject</a> Adresse de l'objet Qt parent
---------------	--

```
00026                                     :QObject (parent),
    temperatureInterieur(0),temperatureExterieur(0),
    horodatageTemperatureInterieur(""),
    horodatageTemperatureExterieur(""),
    heureCouranteTemperatureInterieur(-1),
    heureCouranteTemperatureExterieur(-1)
00027 {
00028 }
```

## 9.23.2.2 ~InfosTemperature()

```
InfosTemperature::~~InfosTemperature ( )
```

```
00031 {
00032 }
```

## 9.23.3 Documentation des fonctions membres

## 9.23.3.1 calculerMoyenneHoraireTemperatureExterieur()

```
void InfosTemperature::calculerMoyenneHoraireTemperatureExterieur ( ) [private]
```

Références [heureCouranteTemperatureExterieur](#), [temperaturesExterieurs](#), et [traitementTemperatureExterieurEnvoye\(\)](#).

Référencé par [traiterNouvelleTemperatureExterieur\(\)](#).

```

00232 {
00233     double sommeTemperatureExterieur = 0;
00234     double temperatureExterieurMoyenne = 0;
00235     double temperatureExterieurMinimum = 999;
00236     double temperatureExterieurMaximum = -999;
00237
00238     // au moins 2 mesures
00239     if(temperaturesExterieurs.size() >= 2)
00240     {
00241         temperatureExterieurMinimum = temperaturesExterieurs[0];
00242         temperatureExterieurMaximum = temperaturesExterieurs[0];
00243         for (int i = 0; i < temperaturesExterieurs.size(); i++)
00244         {
00245             sommeTemperatureExterieur += temperaturesExterieurs[i];
00246
00247             if(temperatureExterieurMinimum > temperaturesExterieurs[i])
00248             {
00249                 temperatureExterieurMinimum = temperaturesExterieurs[i];
00250             }
00251
00252             if(temperatureExterieurMaximum < temperaturesExterieurs[i])
00253             {
00254                 temperatureExterieurMaximum = temperaturesExterieurs[i];
00255             }
00256         }
00257     }
00258     qDebug() << Q_FUNC_INFO << temperaturesExterieurs;
00259     temperatureExterieurMoyenne = sommeTemperatureExterieur / double(temperaturesExterieurs.size());
00260     qDebug() << Q_FUNC_INFO << "temperatureExterieurMoyenne=" << temperatureExterieurMoyenne << "
    temperatureExterieurMinimum=" << temperatureExterieurMinimum << "temperatureExterieurMaximum=" <<
    temperatureExterieurMaximum;
00261     emit traitementTemperatureExterieurEnvoye(
    temperatureExterieurMoyenne, temperatureExterieurMinimum, temperatureExterieurMaximum,
    heureCouranteTemperatureExterieur);
00262     temperaturesExterieurs.clear();
00263 }

```

### 9.23.3.2 calculerMoyenneHoraireTemperatureInterieur()

void InfosTemperature::calculerMoyenneHoraireTemperatureInterieur ( ) [private]

Références [heureCouranteTemperatureInterieur](#), [temperaturesInterieurs](#), et [traitementTemperatureInterieurEnvoye\(\)](#).

Référencé par [traiterMoyenneHoraireTemperatureInterieur\(\)](#).

```

00193 {
00194     double sommeTemperatureInterieur = 0;
00195     double temperatureInterieurMoyenne = 0;
00196     double temperatureInterieurMinimum = 999;
00197     double temperatureInterieurMaximum = -999;
00198
00199     // au moins 2 mesures
00200     if(temperaturesInterieurs.size() >= 2)
00201     {
00202         temperatureInterieurMinimum = temperaturesInterieurs[0];
00203         temperatureInterieurMaximum = temperaturesInterieurs[0];
00204         for (int i = 0; i < temperaturesInterieurs.size(); i++)
00205         {
00206             sommeTemperatureInterieur += temperaturesInterieurs[i];
00207
00208             if(temperatureInterieurMinimum > temperaturesInterieurs[i])
00209             {
00210                 temperatureInterieurMinimum = temperaturesInterieurs[i];
00211             }
00212
00213             if(temperatureInterieurMaximum < temperaturesInterieurs[i])
00214             {
00215                 temperatureInterieurMaximum = temperaturesInterieurs[i];
00216             }
00217         }
00218     }
00219     qDebug() << Q_FUNC_INFO << temperaturesInterieurs;
00220     temperatureInterieurMoyenne = sommeTemperatureInterieur / double(temperaturesInterieurs.size());
00221     qDebug() << Q_FUNC_INFO << "temperatureInterieurMoyenne=" << temperatureInterieurMoyenne << "
    temperatureInterieurMinimum=" << temperatureInterieurMinimum << "temperatureInterieurMaximum=" <<
    temperatureInterieurMaximum;
00222     emit traitementTemperatureInterieurEnvoye(
    temperatureInterieurMoyenne, temperatureInterieurMinimum, temperatureInterieurMaximum,
    heureCouranteTemperatureInterieur);
00223     temperaturesInterieurs.clear();
00224 }

```

### 9.23.3.3 getHorodatageTemperatureExterieur()

```
QString InfosTemperature::getHorodatageTemperatureExterieur ( )
```

Références [horodatageTemperatureExterieur](#).

Référencé par [Ruche : :insererDonneesPortMesureEnvironnement\(\)](#).

```
00107 {  
00108     return horodatageTemperatureExterieur;  
00109 }
```

### 9.23.3.4 getHorodatageTemperatureInterieur()

```
QString InfosTemperature::getHorodatageTemperatureInterieur ( )
```

**Renvoie**

Un *QString* correspondant a l'horodatage temperature intérieur

Références [horodatageTemperatureInterieur](#).

Référencé par [Ruche : :insererDonneesPortMesureRuche\(\)](#).

```
00063 {  
00064     return horodatageTemperatureInterieur;  
00065 }
```

### 9.23.3.5 getTemperatureExterieur()

```
double InfosTemperature::getTemperatureExterieur ( )
```

**Renvoie**

Un *double* correspondant a la temperature exterieur

Références [temperatureExterieur](#).

Référencé par [Alertes : :alertesTemperatureExterieur\(\)](#).

```
00085 {  
00086     return temperatureExterieur;  
00087 }
```

### 9.23.3.6 getTemperatureInterieur()

```
double InfosTemperature::getTemperatureInterieur ( )
```

**Renvoie**

Un *double* correspondant a la temperature intérieur

Références [temperatureInterieur](#).

Référencé par [Alertes : :alertesTemperatureInterieur\(\)](#), et [Ruche : :insererDonneesPortMesureRuche\(\)](#).

```
00041 {  
00042     return temperatureInterieur;  
00043 }
```

### 9.23.3.7 setHorodatageTemperatureExterieur()

```
void InfosTemperature::setHorodatageTemperatureExterieur (  
    const QString horodatageTemperatureExterieur )
```

**Paramètres**

<i>horodatageTemperatureExterieur</i>	correspondant l'attribut <i>horodatageTemperatureExterieur</i>
---------------------------------------	--

Références [horodatageTemperatureExterieur](#).

```
00118 {  
00119     this->horodatageTemperatureExterieur =  
        horodatageTemperatureExterieur;  
00120 }
```

**9.23.3.8 setHorodatageTemperatureInterieur()**

```
void InfosTemperature::setHorodatageTemperatureInterieur (  
    const QString horodatageTemperatureInterieur )
```

Références [horodatageTemperatureInterieur](#).

```
00074 {  
00075     this->horodatageTemperatureInterieur =  
        horodatageTemperatureInterieur;  
00076 }
```

**9.23.3.9 setTemperatureExterieur()**

```
void InfosTemperature::setTemperatureExterieur (  
    double temperatureExterieur )
```

**Paramètres**

<i>temperatureExterieur</i>	correspondant l'attribut <i>TemperatureExterieur</i>
-----------------------------	--

Références [temperatureExterieur](#).

```
00096 {  
00097     this->temperatureExterieur = temperatureExterieur;  
00098 }
```

**9.23.3.10 setTemperatureInterieur()**

```
void InfosTemperature::setTemperatureInterieur (  
    double temperatureInterieur )
```

Références [temperatureInterieur](#).

```
00052 {  
00053     this->temperatureInterieur = temperatureInterieur;  
00054 }
```

### 9.23.3.11 temperatureExterieurEnvoye

```
void InfosTemperature::temperatureExterieurEnvoye (
    const double temperatureExterieur,
    QString horodatage ) [signal]
```

Référencé par [traiterNouvelleTemperatureExterieur\(\)](#).

### 9.23.3.12 temperatureInterieurEnvoye

```
void InfosTemperature::temperatureInterieurEnvoye (
    const double temperatureInterieur,
    QString horodatage ) [signal]
```

Référencé par [traiterNouvelleTemperatureInterieur\(\)](#).

### 9.23.3.13 traitementTemperatureExterieurEnvoye

```
void InfosTemperature::traitementTemperatureExterieurEnvoye (
    const double temperatureExterieurMoyenne,
    const double temperatureExterieurMinimum,
    const double temperatureExterieurMaximum,
    int heure ) [signal]
```

Référencé par [calculerMoyenneHoraireTemperatureExterieur\(\)](#).

### 9.23.3.14 traitementTemperatureInterieurEnvoye

```
void InfosTemperature::traitementTemperatureInterieurEnvoye (
    const double temperatureInterieurMoyenne,
    const double temperatureInterieurMinimum,
    const double temperatureInterieurMaximum,
    int heure ) [signal]
```

Référencé par [calculerMoyenneHoraireTemperatureInterieur\(\)](#).

### 9.23.3.15 traiterMoyenneHoraireTemperatureInterieur()

```
void InfosTemperature::traiterMoyenneHoraireTemperatureInterieur (
    QString horodatageTemperatureInterieur ) [private]
```

#### Paramètres

<i>horodatageTemperatureInterieur</i>	correspondant a l'horodatage de la mesure temperature interieur
---------------------------------------	---

Références [calculerMoyenneHoraireTemperatureInterieur\(\)](#), [heureCouranteTemperatureInterieur](#), [temperatureInterieur](#), et [temperaturesInterieurs](#).

Référencé par [traiterNouvelleTemperatureInterieur\(\)](#).

```

00128 {
00129     QDateTime dateTimeHorodatage = QDateTime::fromString(
00130         horodatageTemperatureInterieur, "dd/MM/yyyy HH:mm:ss");
00131     qDebug() << Q_FUNC_INFO << "heureCouranteTemperatureInterieur" <<
00132         heureCouranteTemperatureInterieur << dateTimeHorodatage.time().hour();
00133     if(heureCouranteTemperatureInterieur == -1)
00134     {
00135         heureCouranteTemperatureInterieur = dateTimeHorodatage.time().hour
00136     );
00137     }
00138     if(heureCouranteTemperatureInterieur == dateTimeHorodatage.time().hour
00139     ())
00140     {
00141         temperaturesInterieurs.append(temperatureInterieur);
00142     }
00143     else if((heureCouranteTemperatureInterieur+1)%24 == dateTimeHorodatage
00144     .time().hour())
00145     {
00146         calculerMoyenneHoraireTemperatureInterieur();
00147         heureCouranteTemperatureInterieur = dateTimeHorodatage.time().hour
00148     );
00149     }
00150     temperaturesInterieurs.append(temperatureInterieur);
00151 }

```

### 9.23.3.16 traiterNouvelleTemperatureExterieur

```

void InfosTemperature::traiterNouvelleTemperatureExterieur (
    QString temperatureExterieurString,
    QString horodatageTemperatureExterieur ) [slot]

```

#### Paramètres

<i>temperatureExterieurString</i>	correspondant à la mesure temperature exterieur
<i>horodatageTemperatureExterieur</i>	correspondant a l'horodatage de la mesure temperature exterieur

Références [calculerMoyenneHoraireTemperatureExterieur\(\)](#), [heureCouranteTemperatureExterieur](#), [horodatageTemperatureExterieur](#), [temperatureExterieur](#), [temperatureExterieurEnvoye\(\)](#), et [temperaturesExterieurs](#).

```

00165 {
00166     temperatureExterieur = temperatureExterieurString.toDouble();
00167     this->horodatageTemperatureExterieur =
00168         horodatageTemperatureExterieur;
00169     emit temperatureExterieurEnvoye(
00170         temperatureExterieur, horodatageTemperatureExterieur);
00171     QDateTime dateTimeHorodatage = QDateTime::fromString(
00172         horodatageTemperatureExterieur, "dd/MM/yyyy HH:mm:ss");
00173     qDebug() << Q_FUNC_INFO << "heureCouranteTemperatureExterieur" <<
00174         heureCouranteTemperatureExterieur << dateTimeHorodatage.time().hour();
00175     if(heureCouranteTemperatureExterieur == -1)
00176     {
00177         heureCouranteTemperatureExterieur = dateTimeHorodatage.time().hour
00178     );
00179     }
00180     if(heureCouranteTemperatureExterieur == dateTimeHorodatage.time().hour
00181     ())
00182     {
00183         temperaturesExterieurs.append(temperatureExterieur);
00184     }
00185     else if((heureCouranteTemperatureExterieur+1)%24 == dateTimeHorodatage
00186     .time().hour())
00187     {
00188         calculerMoyenneHoraireTemperatureExterieur();
00189         heureCouranteTemperatureExterieur = dateTimeHorodatage.time().hour
00190     );
00191     }
00192     temperaturesExterieurs.append(temperatureExterieur);
00193 }

```



### 9.23.3.17 traiterNouvelleTemperatureInterieur

```
void InfosTemperature::traiterNouvelleTemperatureInterieur (
    QString temperatureInterieur,
    QString horodatageTemperatureInterieur ) [slot]
```

Références [horodatageTemperatureInterieur](#), [temperatureInterieur](#), [temperatureInterieurEnvoye\(\)](#), et [traiterMoyenneHoraireTemperatureInterieur\(\)](#).

```
00148 {
00149     qDebug() << Q_FUNC_INFO << "temperatureInterieurString" << temperatureInterieurString <<
        horodatageTemperatureInterieur;
00150     this->horodatageTemperatureInterieur = horodatageTemperatureInterieur;
00151     temperatureInterieur = temperatureInterieurString.toDouble();
00152     emit temperatureInterieurEnvoye(
        temperatureInterieur, horodatageTemperatureInterieur);
00153
00154     traiterMoyenneHoraireTemperatureInterieur(
        horodatageTemperatureInterieur);
00155 }
```

## 9.23.4 Documentation des données membres

### 9.23.4.1 heureCouranteTemperatureExterieur

```
int InfosTemperature::heureCouranteTemperatureExterieur [private]
```

Référencé par [calculerMoyenneHoraireTemperatureExterieur\(\)](#), et [traiterNouvelleTemperatureExterieur\(\)](#).

### 9.23.4.2 heureCouranteTemperatureInterieur

```
int InfosTemperature::heureCouranteTemperatureInterieur [private]
```

Référencé par [calculerMoyenneHoraireTemperatureInterieur\(\)](#), et [traiterMoyenneHoraireTemperatureInterieur\(\)](#).

### 9.23.4.3 horodatageTemperatureExterieur

```
QString InfosTemperature::horodatageTemperatureExterieur [private]
```

Référencé par [getHorodatageTemperatureExterieur\(\)](#), [setHorodatageTemperatureExterieur\(\)](#), et [traiterNouvelleTemperatureExterieur\(\)](#).

### 9.23.4.4 horodatageTemperatureInterieur

```
QString InfosTemperature::horodatageTemperatureInterieur [private]
```

Référencé par [getHorodatageTemperatureInterieur\(\)](#), [setHorodatageTemperatureInterieur\(\)](#), et [traiterNouvelleTemperatureInterieur\(\)](#).

#### 9.23.4.5 `temperatureExterieur`

```
double InfosTemperature::temperatureExterieur [private]
```

Référencé par [getTemperatureExterieur\(\)](#), [setTemperatureExterieur\(\)](#), et [traiterNouvelleTemperatureExterieur\(\)](#).

#### 9.23.4.6 `temperatureInterieur`

```
double InfosTemperature::temperatureInterieur [private]
```

Référencé par [getTemperatureInterieur\(\)](#), [setTemperatureInterieur\(\)](#), [traiterMoyenneHoraireTemperatureInterieur\(\)](#), et [traiterNouvelleTemperatureInterieur\(\)](#).

#### 9.23.4.7 `temperaturesExterieurs`

```
QVector<double> InfosTemperature::temperaturesExterieurs [private]
```

Référencé par [calculerMoyenneHoraireTemperatureExterieur\(\)](#), et [traiterNouvelleTemperatureExterieur\(\)](#).

#### 9.23.4.8 `temperaturesInterieurs`

```
QVector<double> InfosTemperature::temperaturesInterieurs [private]
```

Référencé par [calculerMoyenneHoraireTemperatureInterieur\(\)](#), et [traiterMoyenneHoraireTemperatureInterieur\(\)](#).

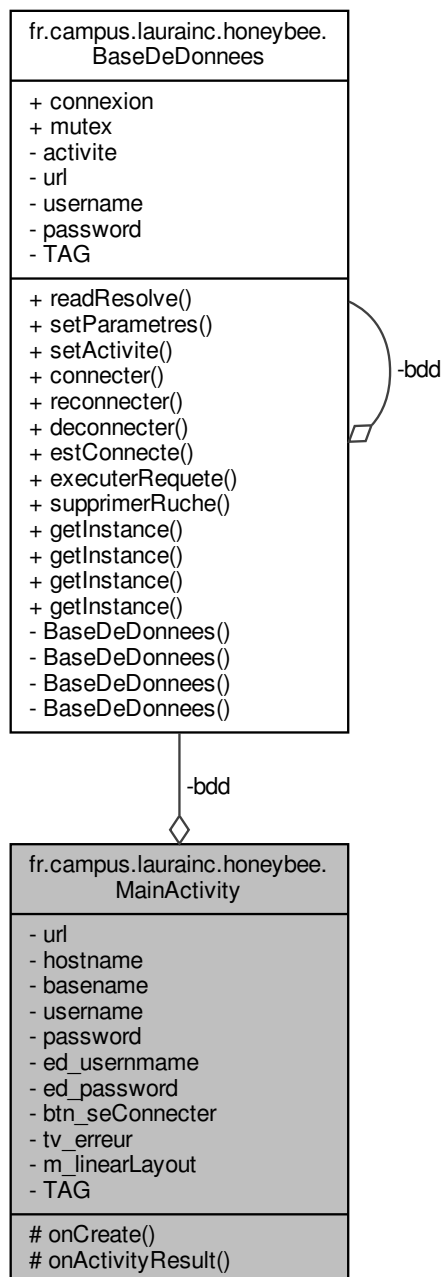
La documentation de cette classe a été générée à partir des fichiers suivants :

- [infosTemperature.h](#)
- [infosHumidite.cpp](#)
- [infosPressionAtmospherique.cpp](#)
- [infosTemperature.cpp](#)

### 9.24 Référence de la classe `fr.campus.laurainc.honeybee.MainActivity`

Activité principale de l'application (Thread UI)

Graphe de collaboration de fr.campus.laurainc.honeybee.MainActivity :



#### Fonctions membres protégées

- void [onCreate](#) (Bundle savedInstanceState)
- void [onActivityResult](#) (int requestCode, int resultCode, Intent intent)

#### Attributs privés

- [BaseDeDonnees bdd](#) = null  
*l'objet permettant un accès à la base de données MySQL*

- String `url`  
*l'URL pointant sur la base de données d'un serveur MySQL*
- String `hostname` = "192.168.52.119"  
*l'adresse IP du serveur MySQL*
- String `basename` = "ruches"  
*le nom de la base de données du serveur MySQL*
- String `username` = "fmellah"  
*le nom du compte utilisateur (root par défaut)*
- String `password` = "password"  
*le mot de passe du compte utilisateur (password par défaut)*
- EditText `ed_username`
- EditText `ed_password`
- Button `btn_seConnecter`
- TextView `tv_erreur`
- LinearLayout `m_linearLayout`

#### Attributs privés statiques

- static final String `TAG` = "MainActivity"  
*le TAG de la classe pour les logs*

#### 9.24.1 Description détaillée

##### Auteur

Clément Laurain

#### 9.24.2 Documentation des fonctions membres

##### 9.24.2.1 onActivityResult()

```
fr.campus.laurainc.honeybee.MainActivity.onActivityResult (
    int requestCode,
    int resultCode,
    Intent intent ) [protected]
```

#### A faire Gestion d'autres paramètres ? MQTT ? ...

Références `fr.campus.laurainc.honeybee.MainActivity.basename`, `fr.campus.laurainc.honeybee.HoneyBee.ID_Intent_Parametres` ← `HoneyBee`, `fr.campus.laurainc.honeybee.BaseDeDonnees.reconnecter()`, et `fr.campus.laurainc.honeybee.BaseDeDonnees.set` ← `Parametres()`.

```
00096    {
00097        //Log.d(TAG, "requestCode=" + requestCode);
00098        //Log.d(TAG, "resultCode=" + resultCode);
00099        if (requestCode == HoneyBee.ID_Intent_ParametresHoneyBee)
00100        {
00101            switch(resultCode)
00102            {
00103                case RESULT_CANCELED:
00104                    // rien à faire ?
00105                    break;
00106                case RESULT_OK:
00107                    // Récupère les paramètres
00108                    hostname = intent.getStringExtra("hostname");
00109                    basename = intent.getStringExtra("basename");
00110                    username = intent.getStringExtra("username");
00111                    password = intent.getStringExtra("password");
00112                    // Recrée l'URL JDBC MySQL
00113                    url = "jdbc:mysql://" + hostname + "/" + basename;
00114                    Log.v(TAG, "url=" + url);
00115                    Log.v(TAG, "username=" + username);
00116                    Log.v(TAG, "password=" + password);
00117                    bdd.setParametres(url, username,
00118                                password);
00119                    bdd.reconnecter();
00120                break;
00121            }
00122        }
00123    }
```

## 9.24.2.2 onCreate()

```
void fr.campus.laurainc.honeybee.MainActivity.onCreate (
    Bundle savedInstanceState ) [protected]
```

**A faire** Affichage de l'état de connexion MySQL ?

Références [fr.campus.laurainc.honeybee.MainActivity.basename](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#).

```
00040     {
00041         super.onCreate(savedInstanceState);
00042         setContentView(R.layout.activity_main);
00043
00044         ed_username = findViewById(R.id.ed_username);
00045         ed_password = findViewById(R.id.ed_password);
00046         btn_seConnecter = findViewById(R.id.btn_seConnecter);
00047         tv_erreur = findViewById(R.id.tv_erreur);
00048
00049         // Initialise l'url pour la connexion à la base de données MySQL
00050         url = "jdbc:mysql://" + hostname + "/" + basename;
00051         // Récupère l'instance de BaseDeDonnees
00052         bdd = BaseDeDonnees.getInstance(this, url,
username, password);
00053         bdd.connecter();
00054
00055         // Fenêtre d'accès aux ruches
00056         final Intent homeActivity = new Intent(MainActivity.this, homeActivity.class);
00057         btn_seConnecter.setOnClickListener(new View.OnClickListener() {
00058             @Override
00059             public void onClick(View v) {
00060                 if(ed_username.getText().toString().equals(username) &&
ed_password.getText().toString().equals(password))
00061                 {
00062                     startActivity(homeActivity);
00063                 }
00064                 else
00065                 {
00066                     tv_erreur.setVisibility(View.VISIBLE);
00067                 }
00068             }
00069         });
00070
00071         // Fenêtre de paramétrage de l'application
00072         /*FloatingActionButton btn_Parametres = findViewById(R.id.btn_Parametres);
00073
00074         btn_Parametres.setOnClickListener(new View.OnClickListener() {
00075
00076             public void onClick(View v) {
00077
00078                 // Crée et démarre une activité
00079
00080                 Intent intent = new Intent(MainActivity.this, ParametresHoneyBeeActivity.class);
00081
00082                 // Passage de données
00083
00084                 intent.putExtra("hostname", hostname);
00085                 intent.putExtra("basename", basename);
00086                 intent.putExtra("username", username);
00087                 intent.putExtra("password", password);
00088
00089                 //startActivity(intent);
00090                 startActivityForResult(intent, HoneyBee.ID_Intent_ParametresHoneyBee);
00091             }
00092         });*/
00093     }
```

## 9.24.3 Documentation des données membres

#### 9.24.3.1 basename

```
String fr.campus.laurainc.honeybee.MainActivity.basename = "ruches" [private]
```

Référencé par [fr.campus.laurainc.honeybee.MainActivity.onActivityResult\(\)](#), et [fr.campus.laurainc.honeybee.MainActivity.onCreate\(\)](#).

#### 9.24.3.2 bdd

```
BaseDeDonnees fr.campus.laurainc.honeybee.MainActivity.bdd = null [private]
```

#### 9.24.3.3 btn\_seConnecter

```
Button fr.campus.laurainc.honeybee.MainActivity.btn_seConnecter [private]
```

#### 9.24.3.4 ed\_password

```
EditText fr.campus.laurainc.honeybee.MainActivity.ed_password [private]
```

#### 9.24.3.5 ed\_username

```
EditText fr.campus.laurainc.honeybee.MainActivity.ed_username [private]
```

#### 9.24.3.6 hostname

```
String fr.campus.laurainc.honeybee.MainActivity.hostname = "192.168.52.119" [private]
```

#### 9.24.3.7 m\_linearLayout

```
LinearLayout fr.campus.laurainc.honeybee.MainActivity.m_linearLayout [private]
```

#### 9.24.3.8 password

```
String fr.campus.laurainc.honeybee.MainActivity.password = "password" [private]
```

#### 9.24.3.9 TAG

```
final String fr.campus.laurainc.honeybee.MainActivity.TAG = "MainActivity" [static], [private]
```

## 9.24.3.10 tv\_erreur

```
TextView fr.campus.laurainc.honeybee.MainActivity.tv_erreur [private]
```

## 9.24.3.11 url

```
String fr.campus.laurainc.honeybee.MainActivity.url [private]
```

## 9.24.3.12 username

```
String fr.campus.laurainc.honeybee.MainActivity.username = "fmellah" [private]
```

La documentation de cette classe a été générée à partir du fichier suivant :

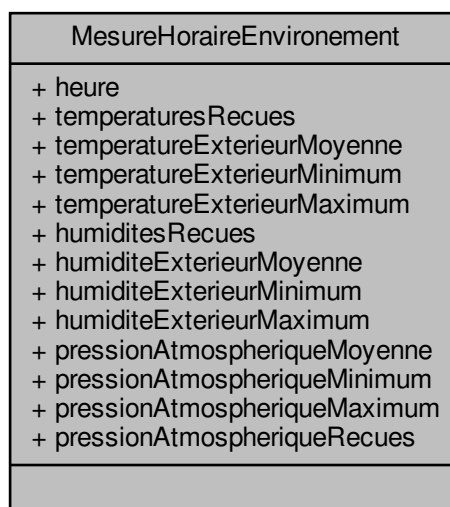
— [MainActivity.java](#)

## 9.25 Référence de la structure MesureHoraireEnvironnement

structure de données pour les mesures horaires

```
#include <ruche.h>
```

Graphe de collaboration de MesureHoraireEnvironnement :



**Attributs publics**

- int [heure](#)
- bool [temperaturesRecues](#) = false
- double [temperatureExterieurMoyenne](#)
- double [temperatureExterieurMinimum](#)
- double [temperatureExterieurMaximum](#)
- bool [humiditesRecues](#) = false
- double [humiditeExterieurMoyenne](#)
- double [humiditeExterieurMinimum](#)
- double [humiditeExterieurMaximum](#)
- double [pressionAtmospheriqueMoyenne](#)
- double [pressionAtmospheriqueMinimum](#)
- double [pressionAtmospheriqueMaximum](#)
- bool [pressionAtmospheriqueRecues](#) = false

**9.25.1 Description détaillée****Auteur**

Florentin Mellah, Enzo Rossi

**Version**

1.1

**9.25.2 Documentation des données membres****9.25.2.1 heure**

```
int MesureHoraireEnvironnement::heure
```

Référencé par [Ruche : :insererMesureHoraireEnvironnement\(\)](#), [Ruche : :recevoirTraitementHumiditeExterieur\(\)](#), [Ruche : :recevoir↵ TraitementPressionAtmospherique\(\)](#), et [Ruche : :recevoirTraitementTemperatureExterieur\(\)](#).

**9.25.2.2 humiditeExterieurMaximum**

```
double MesureHoraireEnvironnement::humiditeExterieurMaximum
```

Référencé par [Ruche : :insererMesureHoraireEnvironnement\(\)](#), et [Ruche : :recevoirTraitementHumiditeExterieur\(\)](#).

**9.25.2.3 humiditeExterieurMinimum**

```
double MesureHoraireEnvironnement::humiditeExterieurMinimum
```

Référencé par [Ruche : :insererMesureHoraireEnvironnement\(\)](#), et [Ruche : :recevoirTraitementHumiditeExterieur\(\)](#).



#### 9.25.2.4 `humiditeExterieurMoyenne`

```
double MesureHoraireEnvironnement::humiditeExterieurMoyenne
```

Référencé par `Ruche : :insererMesureHoraireEnvironnement()`, et `Ruche : :recevoirTraitementHumiditeExterieur()`.

#### 9.25.2.5 `humiditesRecues`

```
bool MesureHoraireEnvironnement::humiditesRecues = false
```

Référencé par `Ruche : :insererMesureHoraireEnvironnement()`, et `Ruche : :recevoirTraitementHumiditeExterieur()`.

#### 9.25.2.6 `pressionAtmospheriqueMaximum`

```
double MesureHoraireEnvironnement::pressionAtmospheriqueMaximum
```

Référencé par `Ruche : :insererMesureHoraireEnvironnement()`, et `Ruche : :recevoirTraitementPressionAtmospherique()`.

#### 9.25.2.7 `pressionAtmospheriqueMinimum`

```
double MesureHoraireEnvironnement::pressionAtmospheriqueMinimum
```

Référencé par `Ruche : :insererMesureHoraireEnvironnement()`, et `Ruche : :recevoirTraitementPressionAtmospherique()`.

#### 9.25.2.8 `pressionAtmospheriqueMoyenne`

```
double MesureHoraireEnvironnement::pressionAtmospheriqueMoyenne
```

Référencé par `Ruche : :insererMesureHoraireEnvironnement()`, et `Ruche : :recevoirTraitementPressionAtmospherique()`.

#### 9.25.2.9 `pressionAtmospheriqueRecues`

```
bool MesureHoraireEnvironnement::pressionAtmospheriqueRecues = false
```

Référencé par `Ruche : :insererMesureHoraireEnvironnement()`, et `Ruche : :recevoirTraitementPressionAtmospherique()`.

#### 9.25.2.10 `temperatureExterieurMaximum`

```
double MesureHoraireEnvironnement::temperatureExterieurMaximum
```

Référencé par `Ruche : :insererMesureHoraireEnvironnement()`, et `Ruche : :recevoirTraitementTemperatureExterieur()`.

### 9.25.2.11 `temperatureExterieurMinimum`

```
double MesureHoraireEnvironnement::temperatureExterieurMinimum
```

Référencé par [Ruche : `:insérerMesureHoraireEnvironnement\(\)`](#), et [Ruche : `:recevoirTraitementTemperatureExterieur\(\)`](#).

### 9.25.2.12 `temperatureExterieurMoyenne`

```
double MesureHoraireEnvironnement::temperatureExterieurMoyenne
```

Référencé par [Ruche : `:insérerMesureHoraireEnvironnement\(\)`](#), et [Ruche : `:recevoirTraitementTemperatureExterieur\(\)`](#).

### 9.25.2.13 `temperaturesRecues`

```
bool MesureHoraireEnvironnement::temperaturesRecues = false
```

Référencé par [Ruche : `:insérerMesureHoraireEnvironnement\(\)`](#), et [Ruche : `:recevoirTraitementTemperatureExterieur\(\)`](#).

La documentation de cette structure a été générée à partir du fichier suivant :

— [ruche.h](#)

## 9.26 Référence de la structure `MesureHoraireRuche`

structure de données pour les mesures horaires

```
#include <ruche.h>
```

Graphe de collaboration de `MesureHoraireRuche` :

MesureHoraireRuche
<ul style="list-style-type: none"> <li>+ <code>heure</code></li> <li>+ <code>temperaturesRecues</code></li> <li>+ <code>temperatureInterieurMoyenne</code></li> <li>+ <code>temperatureInterieurMinimum</code></li> <li>+ <code>temperatureInterieurMaximum</code></li> <li>+ <code>humiditesRecues</code></li> <li>+ <code>humiditeInterieurMoyenne</code></li> <li>+ <code>humiditeInterieurMinimum</code></li> <li>+ <code>humiditeInterieurMaximum</code></li> </ul>

**Attributs publics**

- int `heure`
- bool `temperaturesRecues` = false
- double `temperatureInterieurMoyenne`
- double `temperatureInterieurMinimum`
- double `temperatureInterieurMaximum`
- bool `humiditesRecues` = false
- double `humiditeInterieurMoyenne`
- double `humiditeInterieurMinimum`
- double `humiditeInterieurMaximum`

**9.26.1 Description détaillée****Auteur**

Florentin Mellah, Enzo Rossi

**Version**

1.1

**9.26.2 Documentation des données membres****9.26.2.1 `heure`**

`int` `MesureHoraireRuche::heure`

Référencé par `Ruche : :insererMesureHoraireRuche()`, `Ruche : :recevoirTraitementHumiditeInterieur()`, et `Ruche : :recevoir↵ TraitementTemperatureInterieur()`.

**9.26.2.2 `humiditeInterieurMaximum`**

`double` `MesureHoraireRuche::humiditeInterieurMaximum`

Référencé par `Ruche : :insererMesureHoraireRuche()`, et `Ruche : :recevoirTraitementHumiditeInterieur()`.

**9.26.2.3 `humiditeInterieurMinimum`**

`double` `MesureHoraireRuche::humiditeInterieurMinimum`

Référencé par `Ruche : :insererMesureHoraireRuche()`, et `Ruche : :recevoirTraitementHumiditeInterieur()`.

**9.26.2.4 `humiditeInterieurMoyenne`**

`double` `MesureHoraireRuche::humiditeInterieurMoyenne`

Référencé par `Ruche : :insererMesureHoraireRuche()`, et `Ruche : :recevoirTraitementHumiditeInterieur()`.

### 9.26.2.5 humiditesRecues

```
bool MesureHoraireRuche::humiditesRecues = false
```

Référencé par [Ruche : :insererMesureHoraireRuche\(\)](#), et [Ruche : :recevoirTraitementHumiditeInterieur\(\)](#).

### 9.26.2.6 temperatureInterieurMaximum

```
double MesureHoraireRuche::temperatureInterieurMaximum
```

Référencé par [Ruche : :insererMesureHoraireRuche\(\)](#), et [Ruche : :recevoirTraitementTemperatureInterieur\(\)](#).

### 9.26.2.7 temperatureInterieurMinimum

```
double MesureHoraireRuche::temperatureInterieurMinimum
```

Référencé par [Ruche : :insererMesureHoraireRuche\(\)](#), et [Ruche : :recevoirTraitementTemperatureInterieur\(\)](#).

### 9.26.2.8 temperatureInterieurMoyenne

```
double MesureHoraireRuche::temperatureInterieurMoyenne
```

Référencé par [Ruche : :insererMesureHoraireRuche\(\)](#), et [Ruche : :recevoirTraitementTemperatureInterieur\(\)](#).

### 9.26.2.9 temperaturesRecues

```
bool MesureHoraireRuche::temperaturesRecues = false
```

Référencé par [Ruche : :insererMesureHoraireRuche\(\)](#), et [Ruche : :recevoirTraitementTemperatureInterieur\(\)](#).

La documentation de cette structure a été générée à partir du fichier suivant :  
[— ruche.h](#)

## 9.27 Référence de la structure MesuresHoraireEnsoleillement

structure de données pour les mesures horaires

```
#include <ruche.h>
```

Graphe de collaboration de MesuresHoraireEnsoleillement :

MesuresHoraireEnsoleillement
+ heure + ensoleillementsMoyen + ensoleillementsMinimum + ensoleillementsMaximum + ensoleillementsRecues

**Attributs publics**

- int [heure](#)
- double [ensoleillementsMoyen](#)
- double [ensoleillementsMinimum](#)
- double [ensoleillementsMaximum](#)
- bool [ensoleillementsRecues](#) = false

**9.27.1 Description détaillée****Auteur**

Florentin Mellah, Enzo Rossi

**Version**

1.1

**9.27.2 Documentation des données membres****9.27.2.1 ensoleillementsMaximum**

double MesuresHoraireEnsoleillement::ensoleillementsMaximum

Référencé par [Ruche : :insererMesureHoraireEnsoleillement\(\)](#), et [Ruche : :recevoirTraitementEnsoleillement\(\)](#).

**9.27.2.2 ensoleillementsMinimum**

double MesuresHoraireEnsoleillement::ensoleillementsMinimum

Référencé par [Ruche : :insererMesureHoraireEnsoleillement\(\)](#), et [Ruche : :recevoirTraitementEnsoleillement\(\)](#).

**9.27.2.3 ensoleillementsMoyen**

double MesuresHoraireEnsoleillement::ensoleillementsMoyen

Référencé par [Ruche : :insererMesureHoraireEnsoleillement\(\)](#), et [Ruche : :recevoirTraitementEnsoleillement\(\)](#).

**9.27.2.4 ensoleillementsRecues**

bool MesuresHoraireEnsoleillement::ensoleillementsRecues = false

Référencé par [Ruche : :insererMesureHoraireEnsoleillement\(\)](#), et [Ruche : :recevoirTraitementEnsoleillement\(\)](#).

## 9.27.2.5 heure

```
int MesuresHoraireEnsoleillement::heure
```

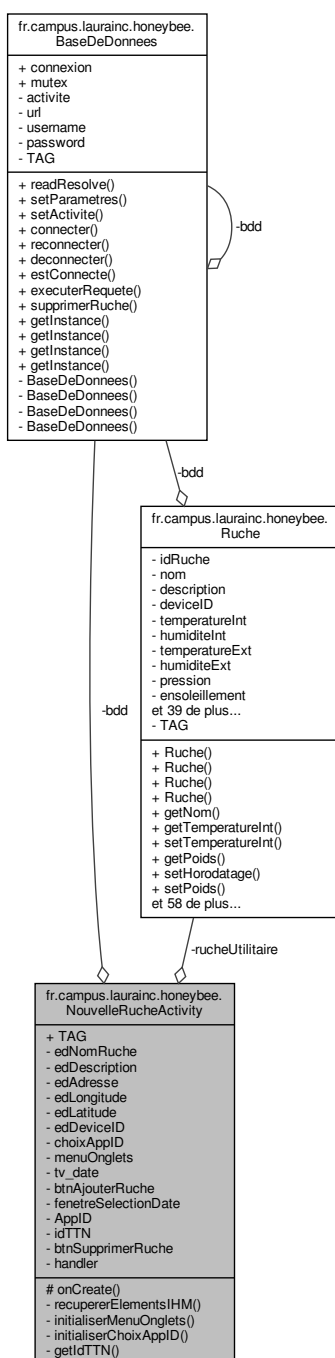
Référencé par [Ruche](#) : `insérerMesureHoraireEnsoleillement()`, et [Ruche](#) : `recevoirTraitementEnsoleillement()`.

La documentation de cette structure a été générée à partir du fichier suivant :

— [ruche.h](#)

## 9.28 Référence de la classe fr.campus.laurainc.honeybee.NouvelleRucheActivity

Graphe de collaboration de fr.campus.laurainc.honeybee.NouvelleRucheActivity :



## Attributs publics statiques

— static final String TAG = "NouvelleRucheActivity"

## Fonctions membres protégées

— void onCreate (Bundle savedInstanceState)

## Fonctions membres privées

— void recupererElementsIHM ()  
 — void initialiserMenuOnglets ()  
 — void initialiserChoixApplID ()  
 — void getIdTTN ()

## Attributs privés

— EditText edNomRuche  
 — EditText edDescription  
 — EditText edAdresse  
 — EditText edLongitude  
 — EditText edLatitude  
 — EditText edDeviceID  
 — Spinner choixApplID  
 — TabHost menuOnglets  
 — TextView tv\_date  
 — Button btnAjouterRuche  
 — DatePickerDialog.OnDateSetListener fenetreSelectionDate  
 — BaseDeDonnees bdd = null  
 — ArrayList< String > ApplID  
 — Ruche rucheUtilitaire  
 — int idTTN  
 — Button btnSupprimerRuche  
 — final Handler handler

## 9.28.1 Documentation des fonctions membres

## 9.28.1.1 getIdTTN()

```
void fr.campus.laurainc.honeybee.NouvelleRucheActivity.getIdTTN ( ) [private]
```

Références [fr.campus.laurainc.honeybee.Ruche.getIdTTNSelectionne\(\)](#).

```
00147    {
00148        idTTN = rucheUtilitaire.getIdTTNSelectionne();
00149        Log.d(TAG, "IdTTN : " + idTTN);
00150    }
```

### 9.28.1.2 initialiserChoixAppID()

```
void fr.campus.laurainc.honeybee.NouvelleRucheActivity.initialiserChoixAppID ( ) [private]
```

Références [fr.campus.laurainc.honeybee.NouvelleRucheActivity.AppID](#), [fr.campus.laurainc.honeybee.Ruche.getListeChoixAppID\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.recupererIdTTN\(\)](#).

```
00121         {
00122             AppID = rucheUtilitaire.getListeChoixAppID();
00123             ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
AppID);
00124             adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
00125             choixAppID.setAdapter(adapter);
00126             choixAppID.setSelection(0);
00127
00128             choixAppID.setOnItemClickListener(new AdapterView.OnItemClickListener()
00129             {
00130                 @Override
00131                 public void onItemClick(AdapterView<?> arg0, View arg1, int position, long id)
00132                 {
00133                     Log.d(TAG, "position : " + position);
00134                     Log.d(TAG, "AppID Sélectionné : " + AppID.get(position));
00135                     rucheUtilitaire.recupererIdTTN(
AppID.get(position));
00136                 }
00137
00138                 @Override
00139                 public void onNothingSelected(AdapterView<?> arg0)
00140                 {
00141                     // TODO Auto-generated method stub
00142                 }
00143             });
00144         }
```

### 9.28.1.3 initialiserMenuOnglets()

```
void fr.campus.laurainc.honeybee.NouvelleRucheActivity.initialiserMenuOnglets ( ) [private]
```

Références [fr.campus.laurainc.honeybee.NouvelleRucheActivity.handler](#), et [fr.campus.laurainc.honeybee.Ruche.recupererChoixChAppID\(\)](#).

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.onCreate\(\)](#).

```
00114     {
00115         rucheUtilitaire = new Ruche(handler);
00116         Log.d(TAG, "Ruche utilitaire créée");
00117         rucheUtilitaire.recupererChoixChAppID();
00118         Log.d(TAG, "Recuperation de la liste des App ID");
00119     }
```

### 9.28.1.4 onCreate()

```
void fr.campus.laurainc.honeybee.NouvelleRucheActivity.onCreate (
    Bundle savedInstanceState ) [protected]
```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.executerRequete\(\)](#), [fr.campus.laurainc.honeybee.NouvelleRucheActivity.fenetreSelectionDate](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#), [fr.campus.laurainc.honeybee.NouvelleRucheActivity.initialiserMenuOnglets\(\)](#), et [fr.campus.laurainc.honeybee.NouvelleRucheActivity.recupererElementsIHM\(\)](#).



```

00050                                     {
00051         super.onCreate(savedInstanceState);
00052         setContentView(R.layout.activity_nouvelle_ruche);
00053
00054         recupererElementsIHM();
00055         initialiserMenuOnglets();
00056
00057         tv_date.setOnClickListener(new View.OnClickListener() {
00058             @Override
00059             public void onClick(View v) {
00060                 Calendar calendrier = Calendar.getInstance();
00061                 int annee = calendrier.get(Calendar.YEAR);
00062                 int mois = calendrier.get(Calendar.MONTH);
00063                 int jour = calendrier.get(Calendar.DAY_OF_MONTH);
00064
00065                 DatePickerDialog dialog = new DatePickerDialog(NouvelleRucheActivity.this, android.R.style.
Theme_DeviceDefault_Light_Dialog, fenetreSelectionDate, annee, mois, jour);
00066
00067                 dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.WHITE));
00068                 dialog.show();
00069             }
00070         });
00071
00072         fenetreSelectionDate = new DatePickerDialog.OnDateSetListener() {
00073             @Override
00074             public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
00075                 month = month + 1;
00076                 Log.d(TAG, "Date sélectionnée " + dayOfMonth + "/" + month + "/" + year );
00077
00078                 String date = year + "-" + month + "-" + dayOfMonth;
00079                 tv_date.setText(date);
00080
00081             }
00082         };
00083
00084         btnAjouterRuche.setOnClickListener(new View.OnClickListener() {
00085             @Override
00086             public void onClick(View v) {
00087                 if (HoneyBee.BDD) {
00088                     bdd = BaseDeDonnees.getInstance();
00089                     bdd.connecter();
00090                     bdd.executerRequete("INSERT INTO Ruche (idTTN, Nom, Description,
DateMiseEnService, Adresse, Longitude, Latitude, DeviceID) VALUES('"+ idTTN + "', '" +
edNomRuche.getText() + "','"+ edDescription.getText() + "','"+
tv_date.getText() + "','"+ edAdresse.getText() + "','"+
edLongitude.getText() + "','"+ edLatitude.getText() + "','"+
edDeviceID.getText() + "')");
00091                     final Intent nouvelleRuche = new Intent(NouvelleRucheActivity.this, DashboardActivity.
class);
00092                     startActivity(nouvelleRuche);
00093                 }
00094             }
00095         });
00096     }
00097 }

```

### 9.28.1.5 recupererElementsIHM()

```
void fr.campus.laurainc.honeybee.NouvelleRucheActivity.recupererElementsIHM ( ) [private]
```

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.onCreate\(\)](#).

```

00100     {
00101         edNomRuche = (EditText) findViewById(R.id.edNomRuche);
00102         edDescription = (EditText) findViewById(R.id.edDescription);
00103         edAdresse = (EditText) findViewById(R.id.edAdresseRuche);
00104         edLongitude = (EditText) findViewById(R.id.edLongitude);
00105         edLatitude = (EditText) findViewById(R.id.edLatitude);
00106         edDeviceID = (EditText) findViewById(R.id.edDeviceID);
00107         tv_date = (TextView) findViewById(R.id.tv_date);
00108         btnAjouterRuche = (Button) findViewById(R.id.btnAjouterRuche);
00109         choixAppID = (Spinner) findViewById(R.id.spinner_AppID);
00110     }
00111 }

```

### 9.28.2 Documentation des données membres

### 9.28.2.1 AppID

```
ArrayList<String> fr.campus.laurainc.honeybee.NouvelleRucheActivity.AppID [private]
```

Référéncé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.initialiserChoixAppID\(\)](#).

### 9.28.2.2 bdd

```
BaseDeDonnees fr.campus.laurainc.honeybee.NouvelleRucheActivity.bdd = null [private]
```

### 9.28.2.3 btnAjouterRuche

```
Button fr.campus.laurainc.honeybee.NouvelleRucheActivity.btnAjouterRuche [private]
```

### 9.28.2.4 btnSupprimerRuche

```
Button fr.campus.laurainc.honeybee.NouvelleRucheActivity.btnSupprimerRuche [private]
```

### 9.28.2.5 choixAppID

```
Spinner fr.campus.laurainc.honeybee.NouvelleRucheActivity.choixAppID [private]
```

### 9.28.2.6 edAdresse

```
EditText fr.campus.laurainc.honeybee.NouvelleRucheActivity.edAdresse [private]
```

### 9.28.2.7 edDescription

```
EditText fr.campus.laurainc.honeybee.NouvelleRucheActivity.edDescription [private]
```

### 9.28.2.8 edDeviceID

```
EditText fr.campus.laurainc.honeybee.NouvelleRucheActivity.edDeviceID [private]
```

### 9.28.2.9 edLatitude

```
EditText fr.campus.laurainc.honeybee.NouvelleRucheActivity.edLatitude [private]
```

### 9.28.2.10 edLongitude

```
EditText fr.campus.laurainc.honeybee.NouvelleRucheActivity.edLongitude [private]
```

### 9.28.2.11 edNomRuche

```
EditText fr.campus.laurainc.honeybee.NouvelleRucheActivity.edNomRuche [private]
```

### 9.28.2.12 fenetreSelectionDate

```
DatePickerDialog.OnDateSetListener fr.campus.laurainc.honeybee.NouvelleRucheActivity.fenetreSelectionDate [private]
```

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.onCreate\(\)](#).

### 9.28.2.13 handler

```
final Handler fr.campus.laurainc.honeybee.NouvelleRucheActivity.handler [private]
```

#### Valeur initiale :

```
= new Handler()
{
    public void handleMessage(Message msg)
    {
        super.handleMessage(msg);
        switch (msg.what)
        {
            case HoneyBee.REQUETE_SQL_ERREUR:
                Log.d(TAG, "handleMessage -> REQUETE SQL ERREUR");
                break;
            case HoneyBee.REQUETE_SQL_LISTE_RUCHES:
                Log.d(TAG, "handleMessage -> REQUETE SQL LISTE RUCHES");
                initialiserChoixAppID();
                break;
            case HoneyBee.REQUETE_SQL_IDTTN:
                Log.d(TAG, "handleMessage -> REQUETE_SQL_IDTTN");
                idTTN = rucheUtilitaire.
                    getIdTTNSelectionne();
                Log.d(TAG, "IdTTN : " + idTTN);
                break;
        }
    }
}
```

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.initialiserMenuOnglets\(\)](#).

### 9.28.2.14 idTTN

```
int fr.campus.laurainc.honeybee.NouvelleRucheActivity.idTTN [private]
```

#### 9.28.2.15 menuOnglets

```
TabHost fr.campus.laurainc.honeybee.NouvelleRucheActivity.menuOnglets [private]
```

#### 9.28.2.16 rucheUtilitaire

```
Ruche fr.campus.laurainc.honeybee.NouvelleRucheActivity.rucheUtilitaire [private]
```

#### 9.28.2.17 TAG

```
final String fr.campus.laurainc.honeybee.NouvelleRucheActivity.TAG = "NouvelleRucheActivity" [static]
```

#### 9.28.2.18 tv\_date

```
TextView fr.campus.laurainc.honeybee.NouvelleRucheActivity.tv_date [private]
```

La documentation de cette classe a été générée à partir du fichier suivant :

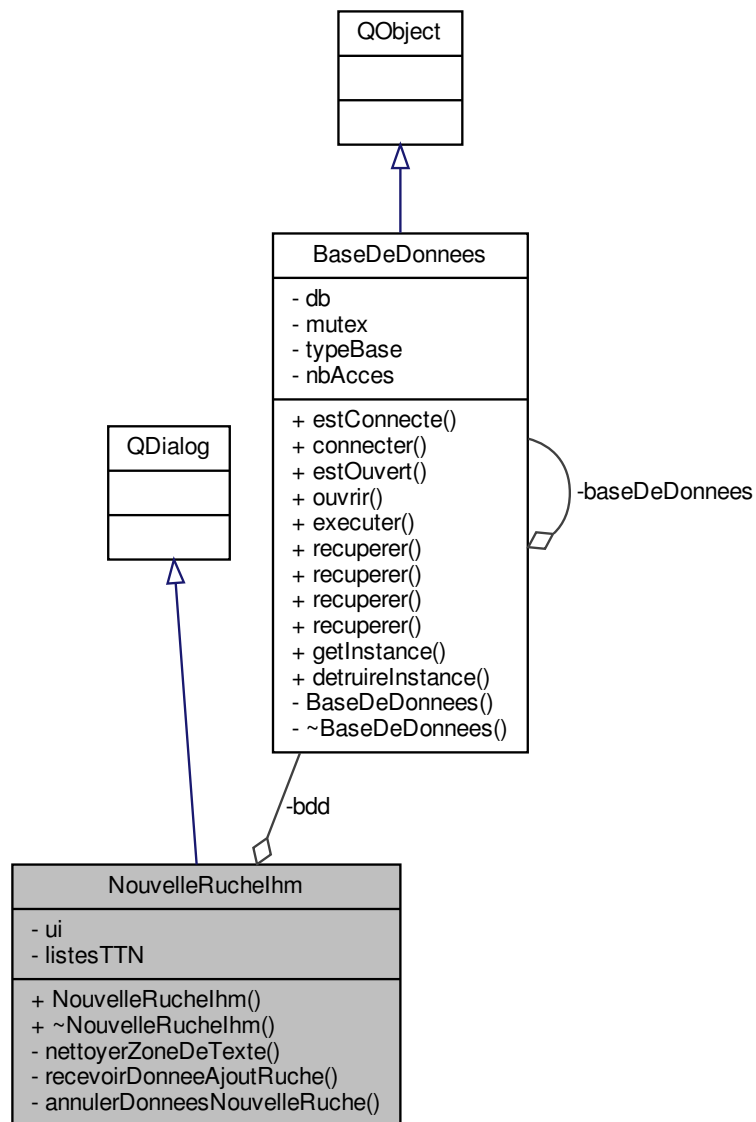
— [NouvelleRucheActivity.java](#)

### 9.29 Référence de la classe NouvelleRucheIhm

Déclaration de la classe [NouvelleRucheIhm](#).

```
#include <nouvelleRucheIhm.h>
```

Graphe de collaboration de NouvelleRuchelhm :



### Fonctions membres publiques

- `NouvelleRuchelhm (QWidget *parent=0)`  
*Constructeur de la fenêtre principale.*
- `~NouvelleRuchelhm ()`  
*Destructeur de la fenêtre principale.*

### Connecteurs privés

- `void recevoirDonneeAjoutRuche ()`  
*slot permettant l'émission des données rentrées par l'utilisateur*
- `void annulerDonneesNouvelleRuche ()`  
*slot permettant grâce à la méthode nettoyerZoneDeTexte d'annuler la saisie des lors que l'on appui sur le bouton annuler*

## Fonctions membres privées

- void `nettoyerZoneDeTexte()`  
*methode qui clear les zones de textes*

## Attributs privés

- `Ui : :NouvelleRucheIhm * ui`
- `BaseDeDonnees * bdd`  
*agrégation de l'objet `BaseDeDonnees`*
- `QVector< QStringList > listesTTN`

## 9.29.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 9.29.2 Documentation des constructeurs et destructeur

## 9.29.2.1 NouvelleRucheIhm()

```
NouvelleRucheIhm::NouvelleRucheIhm (
    QWidget * parent = 0 ) [explicit]
```

## Paramètres

<i>parent</i>	<code>QObject</code> Adresse de l'objet Qt parent (0 = fenêtre principale)
---------------	--

Références `annulerDonneesNouvelleRuche()`, `bdd`, `BDD_NOMBASE`, `BDD_PASSWORD`, `BDD_SERVEUR`, `BDD_USERNAME`, `BaseDeDonnees : :connecter()`, `BaseDeDonnees : :estConnecte()`, `BaseDeDonnees : :getInstance()`, `listesTTN`, `recevoirDonnee` ← `AjoutRuche()`, `BaseDeDonnees : :recuperer()`, et `ui`.

```
00025                                     :
00026         QDialog(parent),
00027         ui(new Ui::NouvelleRucheIhm)
00028 {
00029     qDebug() << Q_FUNC_INFO;
00030     ui->setupUi(this);
00031     setWindowTitle("Ruche 2019 - Création d'une ruche");
00032
00033     bdd = BaseDeDonnees::getInstance();
00034     if(!bdd->estConnecte())
00035         bdd->connecter(BDD_NOMBASE, BDD_USERNAME,
00036                       BDD_PASSWORD, BDD_SERVEUR);
00037
00038     if(bdd->estConnecte())
00039     {
00040         // Une ruche correspond à un DeviceId attaché à un ApplicationID dans le réseau TTN
00041         // récupération des ApplicationID disponibles
00042         QString requeteRecuperationIdTTN = "SELECT idTTN, ApplicationID FROM TTN";
00043         bool retourRequeteRecuperationIdTTN = bdd->recuperer(requeteRecuperationIdTTN,
00044         listesTTN);
00043         if(retourRequeteRecuperationIdTTN)
00044         {
```

```

00045         for(int i=0;i<listesTTN.size();i++)
00046         {
00047             ui->comboBoxListeAppID->addItem(listesTTN.at(i).at(1));
00048         }
00049     }
00050     else
00051         ui->comboBoxListeAppID->addItem("");
00052 }
00053
00054 QDate aujourd'hui = QDate::currentDate();
00055 ui->dateEditMiseEnService->setDate(aujourd'hui);
00056
00057 connect(ui->boutonOk,SIGNAL(clicked(bool)),this,SLOT(
recevoirDonneeAjoutRuche()));
00058 connect(ui->boutonAnnuler,SIGNAL(clicked(bool)),this,SLOT(
annulerDonneesNouvelleRuche()));
00059 }

```

### 9.29.2.2 ~NouvelleRucheIhm()

NouvelleRucheIhm::~~NouvelleRucheIhm ( )

Références [BaseDeDonnees : :destruireInstance\(\)](#), et [ui](#).

```

00068 {
00069     delete ui;
00070     BaseDeDonnees::destruireInstance();
00071 }

```

## 9.29.3 Documentation des fonctions membres

### 9.29.3.1 annulerDonneesNouvelleRuche

void NouvelleRucheIhm::annulerDonneesNouvelleRuche ( ) [private], [slot]

Références [nettoyerZoneDeTexte\(\)](#).

Référencé par [NouvelleRucheIhm\(\)](#).

```

00113 {
00114     nettoyerZoneDeTexte();
00115     reject();
00116 }

```

### 9.29.3.2 nettoyerZoneDeTexte()

void NouvelleRucheIhm::nettoyerZoneDeTexte ( ) [private]

Références [ui](#).

Référencé par [annulerDonneesNouvelleRuche\(\)](#), et [recevoirDonneeAjoutRuche\(\)](#).

```

00124 {
00125     ui->zoneDeTexteNom->clear();
00126     ui->zoneDeTexteDescription->clear();
00127     ui->zoneDeTexteAdresse->clear();
00128     ui->zoneDeTexteLongitude->clear();
00129     ui->zoneDeTexteLatitude->clear();
00130     ui->zoneDeTexteDeviceID->clear();
00131 }

```

### 9.29.3.3 recevoirDonneeAjoutRuche

```
void NouvelleRucheIhm::recevoirDonneeAjoutRuche ( ) [private], [slot]
```

Références [APP\\_TITRE](#), [bdd](#), [BaseDeDonnees : :executer\(\)](#), [listesTTN](#), [nettoyerZoneDeTexte\(\)](#), [BaseDeDonnees : :recuperer\(\)](#), et [ui](#).

Référencé par [NouvelleRucheIhm\(\)](#).

```
00079 {
00080     if(listesTTN.size() == 0)
00081     {
00082         QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Impossible
d'ajouter la nouvelle ruche : TTN manquant !"));
00083         return;
00084     }
00085
00086     QDate date = QDate::fromString(ui->dateEditMiseEnService->text(), "dd/MM/yyyy");
00087     QString requete = "INSERT INTO Ruche (idTTN, Nom, Description, DateMiseEnService, Adresse, Longitude,
Latitude, DeviceID) VALUES ('" + listesTTN.at(ui->comboBoxListeAppID->currentIndex()).at(0) + " ,
'" + ui->zoneDeTexteNom->text() + "', '" + ui->zoneDeTexteDescription->text() + "', '" + date.toString(
"yyyy-MM-dd") + "', '" + ui->zoneDeTexteAdresse->text() + "', '" + ui->zoneDeTexteLongitude->text() + "
', '" + ui->zoneDeTexteLatitude->text() + "', '" + ui->zoneDeTexteDeviceID->text() + "')";
00088     qDebug() << Q_FUNC_INFO << requete;
00089     bool retour = bdd->executer(requete);
00090     if(retour)
00091     {
00092         QString idRuche;
00093         requete = "SELECT idRuche FROM Ruche WHERE DeviceID='" + ui->zoneDeTexteDeviceID->text() + "'";
00094         retour = bdd->recuperer(requete, idRuche);
00095         if(retour)
00096         {
00097             requete = "INSERT INTO Seuils (idRuche, TemperatureIntMin, TemperatureIntMax, HumiditeIntMin,
HumiditeIntMax, TemperatureExtMin, TemperatureExtMax, HumiditeExtMin, HumiditeExtMax, PressionMin,
PressionMax, PoidsMin, PoidsMax, EnsoleillementMin, EnsoleillementMax, Charge) VALUES ('" + idRuche + " , '" + "25."
+ "', '" + "35.0" + "', '" + "20" + "', '" + "30" + "', '" + "5" + "', '" + "35" + "', '" + "20" + "', '" +
"35" + "', '" + "1000" + "', '" + "1200" + "', '" + "10" + "', '" + "100" + "', '" + "10" + "', '" + "1000"
+ "', '" + "25" + "')";
00098             retour = bdd->executer(requete);
00099         }
00100         nettoyerZoneDeTexte();
00101         accept();
00102     }
00103     else
00104     {
00105         QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Impossible
d'ajouter la nouvelle ruche !"));
00106     }
```

## 9.29.4 Documentation des données membres

### 9.29.4.1 bdd

```
BaseDeDonnees* NouvelleRucheIhm::bdd [private]
```

Référencé par [NouvelleRucheIhm\(\)](#), et [recevoirDonneeAjoutRuche\(\)](#).

### 9.29.4.2 listesTTN

```
QVector<QStringList> NouvelleRucheIhm::listesTTN [private]
```

Référencé par [NouvelleRucheIhm\(\)](#), et [recevoirDonneeAjoutRuche\(\)](#).



## 9.29.4.3 ui

```
Ui::NouvelleRucheIhm* NouvelleRucheIhm::ui [private]
```

Référencé par [nettoyerZoneDeTexte\(\)](#), [NouvelleRucheIhm\(\)](#), [recevoirDonneeAjoutRuche\(\)](#), et [~NouvelleRucheIhm\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

- [nouvelleRucheIhm.h](#)
- [nouvelleRucheIhm.cpp](#)

## 9.30 Référence de la classe fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity

Activité de paramétrage de l'application.

Graphe de collaboration de fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity :

fr.campus.laurainc.honeybee. ParametresHoneyBeeActivity
~ edtHostname ~ edtBasename ~ edtUsername ~ edtPassword - hostname - basename - username - password - TAG
+ valider() + annuler() # onCreate()

## Fonctions membres publiques

- void [valider](#) (View view)  
*Quitte l'activité en appliquant les paramètres.*
- void [annuler](#) (View view)  
*Quitte l'activité sans prendre en compte les paramètres.*

## Fonctions membres protégées

- void [onCreate](#) (Bundle savedInstanceState)

## Attributs privés

- String [hostname](#)  
*l'adress IP du serveur MySQL*
- String [basename](#) = "ruche"  
*le nom de la base de données du serveur MySQL*
- String [username](#) = "root"  
*le nom du compte utilisateur (root par défaut)*
- String [password](#) = "password"  
*le mot de passe du compte utilisateur (password par défaut)*

## Attributs privés statiques

— static final String TAG = "ParametresHoneyBeeActivity"  
 le TAG de la classe pour les logs

## 9.30.1 Description détaillée

**A faire** D'autres paramètres ? MQTT ? ...

## 9.30.2 Documentation des fonctions membres

## 9.30.2.1 annuler()

```
fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity.annuler (
    View view )
```

## Paramètres

<a href="#">view</a>	<a href="#">View la vue associée</a>
----------------------	--------------------------------------

```
00078      {
00079          Intent intent = new Intent();
00080          setResult (RESULT_CANCELED, intent);
00081          finish();
00082      }
```

## 9.30.2.2 onCreate()

```
void fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity.onCreate (
    Bundle savedInstanceState ) [protected]
```

```
00031      {
00032          super.onCreate(savedInstanceState);
00033          setContentView(R.layout.activity_parametres_honey_bee);
00034
00035          Intent intent = getIntent();
00036          hostname = intent.getStringExtra("hostname");
00037          basename = intent.getStringExtra("basename");
00038          username = intent.getStringExtra("username");
00039          password = intent.getStringExtra("password");
00040
00041          edtHostname = (EditText) this.findViewById(R.id.edtHostname);
00042          edtBasename = (EditText) this.findViewById(R.id.edtBasename);
00043          edtUsername = (EditText) this.findViewById(R.id.edtUsername);
00044          edtPassword = (EditText) this.findViewById(R.id.edtPassword);
00045
00046          edtHostname.setText(hostname, TextView.BufferType.EDITABLE);
00047          edtBasename.setText(basename, TextView.BufferType.EDITABLE);
00048          edtUsername.setText(username, TextView.BufferType.EDITABLE);
00049          edtPassword.setText(password, TextView.BufferType.EDITABLE);
00050      }
```

## 9.30.2.3 valider()

```
fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity.valider (
    View view )
```

## Paramètres

<a href="#">view</a>	<a href="#">View la vue associée</a>
----------------------	--------------------------------------

```
00058    {
00059        Intent intent = new Intent();
00060        hostname = edtHostname.getText().toString();
00061        basename = edtBasename.getText().toString();
00062        username = edtUsername.getText().toString();
00063        password = edtPassword.getText().toString();
00064        intent.putExtra("hostname", hostname);
00065        intent.putExtra("basename", basename);
00066        intent.putExtra("username", username);
00067        intent.putExtra("password", password);
00068        setResult(RESULT_OK, intent);
00069        finish();
00070    }
```

## 9.30.3 Documentation des données membres

## 9.30.3.1 basename

```
String fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity.basename = "ruche" [private]
```

## 9.30.3.2 hostname

```
String fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity.hostname [private]
```

## 9.30.3.3 password

```
String fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity.password = "password" [private]
```

## 9.30.3.4 TAG

```
final String fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity.TAG = "ParametresHoneyBeeActivity" [static],
[private]
```

## 9.30.3.5 username

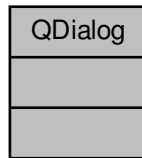
```
String fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity.username = "root" [private]
```

La documentation de cette classe a été générée à partir du fichier suivant :

— [ParametresHoneyBeeActivity.java](#)

### 9.31 Référence de la classe QDialog

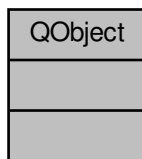
Graphe de collaboration de QDialog :



La documentation de cette classe a été générée à partir du fichier suivant : [reglagesAlertesImm.h](#)

### 9.32 Référence de la classe QObject

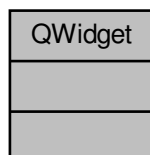
Graphe de collaboration de QObject :



La documentation de cette classe a été générée à partir du fichier suivant : [infosPoids.h](#)

### 9.33 Référence de la classe QWidget

Graphe de collaboration de QWidget :



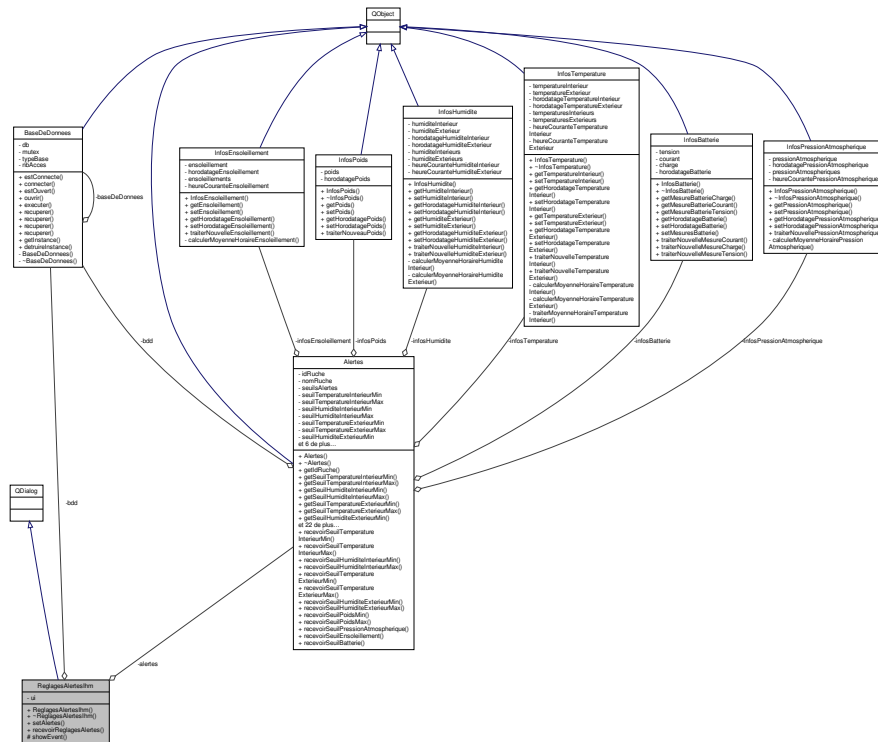
La documentation de cette classe a été générée à partir du fichier suivant : [rucheImm.h](#)

## 9.34 Référence de la classe ReglagesAlertesIhm

Déclaration de la classe [ReglagesAlertesIhm](#).

```
#include <reglagesAlertesIhm.h>
```

Graphe de collaboration de ReglagesAlertesIhm :



## Connecteurs publics

— void [recevoirReglagesAlertes](#) ()  
reçoit les paramètres de l'ihm

## Signaux

— void [envoiSeuilTemperatureInterieurMin](#) (QString seuil)  
— void [envoiSeuilTemperatureInterieurMax](#) (QString seuil)  
— void [envoiSeuilHumiditeInterieurMin](#) (QString seuil)  
— void [envoiSeuilHumiditeInterieurMax](#) (QString seuil)  
— void [envoiSeuilTemperatureExterieurMin](#) (QString seuil)  
— void [envoiSeuilTemperatureExterieurMax](#) (QString seuil)  
— void [envoiSeuilHumiditeExterieurMin](#) (QString seuil)  
— void [envoiSeuilHumiditeExterieurMax](#) (QString seuil)  
— void [envoiSeuilEnsoleillementMin](#) (QString seuil)  
— void [envoiSeuilPressionAtmospheriqueMin](#) (QString seuil)  
— void [envoiSeuilPoidsMin](#) (QString seuil)  
— void [envoiSeuilPoidsMax](#) (QString seuil)

## Fonctions membres publiques

— [ReglagesAlertesIhm](#) (QWidget \*parent=0)  
Constructeur de la fenêtre principale.  
— ~[ReglagesAlertesIhm](#) ()  
Destructeur de la fenêtre principale.  
— void [setAlertes](#) (Alertes \*alertes)  
setter de l'objet alertes

## Fonctions membres protégées

- void [showEvent](#) (QShowEvent \*ev)  
*affiche les seuils à chaque fois que la fenêtre sera affichée*

## Attributs privés

- Ui : :ReglagesAlertesIhm \* [ui](#)
- [Alertes](#) \* [alertes](#)
- [BaseDeDonnees](#) \* [bdd](#)  
*agrégation de l'objet [BaseDeDonnees](#)*

## 9.34.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 9.34.2 Documentation des constructeurs et destructeur

## 9.34.2.1 ReglagesAlertesIhm()

```
ReglagesAlertesIhm::ReglagesAlertesIhm (
    QWidget * parent = 0 ) [explicit]
```

## Paramètres

<i>parent</i>	<a href="#">QObject</a> Adresse de l'objet Qt parent (0 = fenêtre principale)
---------------	---

Références [bdd](#), [BDD\\_NOMBASE](#), [BDD\\_PASSWORD](#), [BDD\\_SERVEUR](#), [BDD\\_USERNAME](#), [BaseDeDonnees](#) : :connecter(), [BaseDeDonnees](#) : :estConnecte(), [BaseDeDonnees](#) : :getInstance(), [recevoirReglagesAlertes\(\)](#), et [ui](#).

```
00025                                     :
00026         QDialog(parent),
00027         ui(new Ui::ReglagesAlertesIhm), alertes(0)
00028 {
00029     ui->setUi(this);
00030
00031     bdd = BaseDeDonnees::getInstance();
00032     if(!bdd->estConnecte())
00033         bdd->connecter(BDD\_NOMBASE, BDD\_USERNAME,
00034                     BDD\_PASSWORD, BDD\_SERVEUR);
00035     connect(ui->pushButtonOk, SIGNAL(clicked(bool)), this, SLOT(accept()));
00036     connect(ui->pushButtonOk, SIGNAL(clicked(bool)), this, SLOT(
00037         recevoirReglagesAlertes()));
00037 }
```

### 9.34.2.2 ~ReglagesAlertesIhm()

ReglagesAlertesIhm::~~ReglagesAlertesIhm ( )

Références [BaseDeDonnees::destruireInstance\(\)](#), et [ui](#).

```
00046 {  
00047     delete ui;  
00048     BaseDeDonnees::destruireInstance();  
00049 }
```

## 9.34.3 Documentation des fonctions membres

### 9.34.3.1 envoiSeuilEnsoleillementMin

```
void ReglagesAlertesIhm::envoiSeuilEnsoleillementMin (  
    QString seuil ) [signal]
```

Référéncé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

### 9.34.3.2 envoiSeuilHumiditeExterieurMax

```
void ReglagesAlertesIhm::envoiSeuilHumiditeExterieurMax (  
    QString seuil ) [signal]
```

Référéncé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

### 9.34.3.3 envoiSeuilHumiditeExterieurMin

```
void ReglagesAlertesIhm::envoiSeuilHumiditeExterieurMin (  
    QString seuil ) [signal]
```

Référéncé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

### 9.34.3.4 envoiSeuilHumiditeInterieurMax

```
void ReglagesAlertesIhm::envoiSeuilHumiditeInterieurMax (  
    QString seuil ) [signal]
```

Référéncé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

#### 9.34.3.5 envoiSeuilHumiditeInterieurMin

```
void ReglagesAlertesIhm::envoiSeuilHumiditeInterieurMin (
    QString seuil ) [signal]
```

Référencé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

#### 9.34.3.6 envoiSeuilPoidsMax

```
void ReglagesAlertesIhm::envoiSeuilPoidsMax (
    QString seuil ) [signal]
```

Référencé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

#### 9.34.3.7 envoiSeuilPoidsMin

```
void ReglagesAlertesIhm::envoiSeuilPoidsMin (
    QString seuil ) [signal]
```

Référencé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

#### 9.34.3.8 envoiSeuilPressionAtmospheriqueMin

```
void ReglagesAlertesIhm::envoiSeuilPressionAtmospheriqueMin (
    QString seuil ) [signal]
```

Référencé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

#### 9.34.3.9 envoiSeuilTemperatureExterieurMax

```
void ReglagesAlertesIhm::envoiSeuilTemperatureExterieurMax (
    QString seuil ) [signal]
```

Référencé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

#### 9.34.3.10 envoiSeuilTemperatureExterieurMin

```
void ReglagesAlertesIhm::envoiSeuilTemperatureExterieurMin (
    QString seuil ) [signal]
```

Référencé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).



## 9.34.3.11 envoiSeuilTemperatureInterieurMax

```
void ReglagesAlertesIhm::envoiSeuilTemperatureInterieurMax (
    QString seuil ) [signal]
```

Référencé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

## 9.34.3.12 envoiSeuilTemperatureInterieurMin

```
void ReglagesAlertesIhm::envoiSeuilTemperatureInterieurMin (
    QString seuil ) [signal]
```

Référencé par [recevoirReglagesAlertes\(\)](#), et [setAlertes\(\)](#).

## 9.34.3.13 recevoirReglagesAlertes

```
void ReglagesAlertesIhm::recevoirReglagesAlertes ( ) [slot]
```

Références [alertes](#), [bdd](#), [envoiSeuilEnsoleillementMin\(\)](#), [envoiSeuilHumiditeExterieurMax\(\)](#), [envoiSeuilHumiditeExterieurMin\(\)](#), [envoiSeuilHumiditeInterieurMax\(\)](#), [envoiSeuilHumiditeInterieurMin\(\)](#), [envoiSeuilPoidsMax\(\)](#), [envoiSeuilPoidsMin\(\)](#), [envoiSeuilPressionAtmospheriqueMin\(\)](#), [envoiSeuilTemperatureExterieurMax\(\)](#), [envoiSeuilTemperatureExterieurMin\(\)](#), [envoiSeuilTemperatureInterieurMax\(\)](#), [envoiSeuilTemperatureInterieurMin\(\)](#), [BaseDeDonnees : :executer\(\)](#), [Alertes : :getIdRuche\(\)](#), [PRESSION\\_ATMOSPHERIQUE\\_SEUIL\\_MAX](#), et [ui](#).

Référencé par [ReglagesAlertesIhm\(\)](#).

```
00107 {
00108     QString temperatureInterieurMin;
00109     QString temperatureInterieurMax;
00110     QString humiditeInterieurMin;
00111     QString humiditeInterieurMax;
00112     QString temperatureExterieurMin;
00113     QString temperatureExterieurMax;
00114     QString humiditeExterieurMin;
00115     QString humiditeExterieurMax;
00116     QString poidsMin;
00117     QString poidsMax;
00118     QString pressionAtmospheriqueMin;
00119     QString ensoleillementMin;
00120
00121     temperatureInterieurMin = ui->lineEditSeuilTemperatureInterieurMin->text();
00122     temperatureInterieurMax = ui->lineEditSeuilTemperatureInterieurMax->text();
00123     humiditeInterieurMin = ui->lineEditSeuilHumiditeInterieurMin->text();
00124     humiditeInterieurMax = ui->lineEditSeuilHumiditeInterieurMax->text();
00125     temperatureExterieurMin = ui->lineEditSeuilTemperatureExterieurMin->text();
00126     temperatureExterieurMax = ui->lineEditSeuilTemperatureExterieurMax->text();
00127     humiditeExterieurMin = ui->lineEditSeuilHumiditeExterieurMin->text();
00128     humiditeExterieurMax = ui->lineEditSeuilHumiditeExterieurMax->text();
00129     poidsMin = ui->lineEditSeuilPoidsMin->text();
00130     poidsMax = ui->lineEditSeuilPoidsMax->text();
00131     pressionAtmospheriqueMin = ui->lineEditSeuilPressionAtmospherique->text();
00132     ensoleillementMin = ui->lineEditSeuilEnsoleillement->text();
00133
00134     QString requete = "UPDATE Seuils SET TemperatureIntMin='" + temperatureInterieurMin + "',
    TemperatureIntMax='" + temperatureInterieurMax + "', HumiditeIntMin='" + humiditeInterieurMin + "', HumiditeIntMax='" +
    humiditeInterieurMax + "', TemperatureExtMin='" + temperatureExterieurMin + "', TemperatureExtMax='" +
    temperatureExterieurMax + "', HumiditeExtMin='" + humiditeExterieurMin + "', HumiditeExtMax='" +
    humiditeExterieurMax + "', PressionMin='" + pressionAtmospheriqueMin + "', PressionMax='" + QString::number(
    PRESSION_ATMOSPHERIQUE_SEUIL_MAX) + "', PoidsMin='" + poidsMin + "',
    PoidsMax='" + poidsMax + "' WHERE idRuche='" + this->alertes->getIdRuche() + "'";
00135     qDebug() << Q_FUNC_INFO << requete;
00136     bool retour = bdd->executer(requete);
00137     if (retour)
00138     {
00139         qDebug() << Q_FUNC_INFO << temperatureInterieurMin;
00140         emit envoiSeuilTemperatureInterieurMin(temperatureInterieurMin);
00141         emit envoiSeuilTemperatureInterieurMax(temperatureInterieurMax);
00142         emit envoiSeuilHumiditeInterieurMin(humiditeInterieurMin);
00143         emit envoiSeuilHumiditeInterieurMax(humiditeInterieurMax);
```

```

00144         emit envoiSeuilTemperatureExterieurMin(temperatureExterieurMin);
00145         emit envoiSeuilTemperatureExterieurMax(temperatureExterieurMax);
00146         emit envoiSeuilHumiditeExterieurMin(humiditeExterieurMin);
00147         emit envoiSeuilHumiditeExterieurMax(humiditeExterieurMax);
00148         emit envoiSeuilPoidsMin(poidsMin);
00149         emit envoiSeuilPoidsMax(poidsMax);
00150         emit envoiSeuilPressionAtmospheriqueMin(pressionAtmospheriqueMin);
00151     };
00152     }
00153 }

```

#### 9.34.3.14 setAlertes()

```

void ReglagesAlertesIhm::setAlertes (
    Alertes * alertes )

```

##### Paramètres

* <i>alertes</i>	correspond à l'objet de la classe <a href="#">Ruchelhm</a>
------------------	--

Références [alertes](#), [envoiSeuilEnsoleillementMin\(\)](#), [envoiSeuilHumiditeExterieurMax\(\)](#), [envoiSeuilHumiditeExterieurMin\(\)](#), [envoiSeuilHumiditeInterieurMax\(\)](#), [envoiSeuilHumiditeInterieurMin\(\)](#), [envoiSeuilPoidsMax\(\)](#), [envoiSeuilPoidsMin\(\)](#), [envoiSeuilPressionAtmospheriqueMin\(\)](#), [envoiSeuilTemperatureExterieurMax\(\)](#), [envoiSeuilTemperatureExterieurMin\(\)](#), [envoiSeuilTemperatureInterieurMax\(\)](#), et [envoiSeuilTemperatureInterieurMin\(\)](#).

Référencé par [Ruchelhm](#) : [:selectionnerRuche\(\)](#).

```

00057 {
00058     if(this->alertes == 0)
00059     {
00060         qDebug() << Q_FUNC_INFO << this->alertes;
00061         this->alertes = alertes;
00062         connect(this, SIGNAL(envoiSeuilTemperatureInterieurMin(QString)),
this->alertes, SLOT(recevoirSeuilTemperatureInterieurMin(QString)));
00063         connect(this, SIGNAL(envoiSeuilTemperatureInterieurMax(QString)),
this->alertes, SLOT(recevoirSeuilTemperatureInterieurMax(QString)));
00064         connect(this, SIGNAL(envoiSeuilHumiditeInterieurMin(QString)), this->
alertes, SLOT(recevoirSeuilHumiditeInterieurMin(QString)));
00065         connect(this, SIGNAL(envoiSeuilHumiditeInterieurMax(QString)), this->
alertes, SLOT(recevoirSeuilHumiditeInterieurMax(QString)));
00066         connect(this, SIGNAL(envoiSeuilTemperatureExterieurMin(QString)),
this->alertes, SLOT(recevoirSeuilTemperatureExterieurMin(QString)));
00067         connect(this, SIGNAL(envoiSeuilTemperatureExterieurMax(QString)),
this->alertes, SLOT(recevoirSeuilTemperatureExterieurMax(QString)));
00068         connect(this, SIGNAL(envoiSeuilHumiditeExterieurMin(QString)), this->
alertes, SLOT(recevoirSeuilHumiditeExterieurMin(QString)));
00069         connect(this, SIGNAL(envoiSeuilHumiditeExterieurMax(QString)), this->
alertes, SLOT(recevoirSeuilHumiditeExterieurMax(QString)));
00070         connect(this, SIGNAL(envoiSeuilPoidsMin(QString)), this->alertes, SLOT(
recevoirSeuilPoidsMin(QString)));
00071         connect(this, SIGNAL(envoiSeuilPoidsMax(QString)), this->alertes, SLOT(
recevoirSeuilPoidsMax(QString)));
00072         connect(this, SIGNAL(envoiSeuilPressionAtmospheriqueMin(QString))
, this->alertes, SLOT(recevoirSeuilPressionAtmospherique(QString)));
00073         connect(this, SIGNAL(envoiSeuilEnsoleillementMin(QString)), this->
alertes, SLOT(recevoirSeuilEnsoleillement(QString)));
00074     }
00075 }

```

#### 9.34.3.15 showEvent()

```

void ReglagesAlertesIhm::showEvent (
    QShowEvent * ev ) [protected]

```

## Paramètres

ev	QShowEvent* l'évènement
----	-------------------------

Références [alertes](#), [Alertes](#) : [:getSeuilEnsoleillement\(\)](#), [Alertes](#) : [:getSeuilHumiditeExterieurMax\(\)](#), [Alertes](#) : [:getSeuilHumiditeExterieurMin\(\)](#), [Alertes](#) : [:getSeuilHumiditeInterieurMax\(\)](#), [Alertes](#) : [:getSeuilHumiditeInterieurMin\(\)](#), [Alertes](#) : [:getSeuilPoidsMax\(\)](#), [Alertes](#) : [:getSeuilPoidsMin\(\)](#), [Alertes](#) : [:getSeuilPressionAtmospherique\(\)](#), [Alertes](#) : [:getSeuilTemperatureExterieurMax\(\)](#), [Alertes](#) : [:getSeuilTemperatureExterieurMin\(\)](#), [Alertes](#) : [:getSeuilTemperatureInterieurMax\(\)](#), [Alertes](#) : [:getSeuilTemperatureInterieurMin\(\)](#), et [ui](#).

```
00083 {
00084     Q_UNUSED(ev)
00085     if (this->alertes != 0)
00086     {
00087         ui->lineEditSeuilTemperatureInterieurMin->setText (QString::number (this->
alertes->getSeuilTemperatureInterieurMin()));
00088         ui->lineEditSeuilTemperatureInterieurMax->setText (QString::number (this->
alertes->getSeuilTemperatureInterieurMax()));
00089         ui->lineEditSeuilHumiditeInterieurMin->setText (QString::number (this->
alertes->getSeuilHumiditeInterieurMin()));
00090         ui->lineEditSeuilHumiditeInterieurMax->setText (QString::number (this->
alertes->getSeuilHumiditeInterieurMax()));
00091         ui->lineEditSeuilTemperatureExterieurMin->setText (QString::number (this->
alertes->getSeuilTemperatureExterieurMin()));
00092         ui->lineEditSeuilTemperatureExterieurMax->setText (QString::number (this->
alertes->getSeuilTemperatureExterieurMax()));
00093         ui->lineEditSeuilHumiditeExterieurMin->setText (QString::number (this->
alertes->getSeuilHumiditeExterieurMin()));
00094         ui->lineEditSeuilHumiditeExterieurMax->setText (QString::number (this->
alertes->getSeuilHumiditeExterieurMax()));
00095         ui->lineEditSeuilPoidsMin->setText (QString::number (this->alertes->
getSeuilPoidsMin()));
00096         ui->lineEditSeuilPoidsMax->setText (QString::number (this->alertes->
getSeuilPoidsMax()));
00097         ui->lineEditSeuilPressionAtmospherique->setText (QString::number (this->
alertes->getSeuilPressionAtmospherique()));
00098         ui->lineEditSeuilEnsoleillement->setText (QString::number (this->alertes->
getSeuilEnsoleillement()));
00099     }
00100 }
```

## 9.34.4 Documentation des données membres

## 9.34.4.1 alertes

[Alertes](#)\* [ReglagesAlertesIhm::alertes](#) [private]

Référencé par [recevoirReglagesAlertes\(\)](#), [setAlertes\(\)](#), et [showEvent\(\)](#).

## 9.34.4.2 bdd

[BaseDeDonnees](#)\* [ReglagesAlertesIhm::bdd](#) [private]

Référencé par [recevoirReglagesAlertes\(\)](#), et [ReglagesAlertesIhm\(\)](#).

## 9.34.4.3 ui

[Ui::ReglagesAlertesIhm](#)\* [ReglagesAlertesIhm::ui](#) [private]

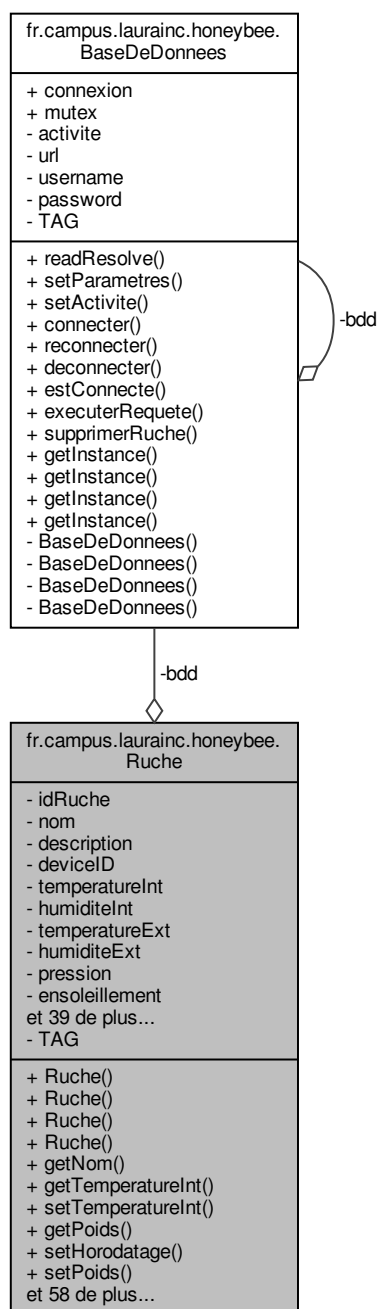
Référencé par [recevoirReglagesAlertes\(\)](#), [ReglagesAlertesIhm\(\)](#), [showEvent\(\)](#), et [~ReglagesAlertesIhm\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

- [reglagesAlertesIhm.h](#)
- [reglagesAlertesIhm.cpp](#)

### 9.35 Référence de la classe fr.campus.laurainc.honeybee.Ruche

Graphe de collaboration de fr.campus.laurainc.honeybee.Ruche :



#### Fonctions membres publiques

- [Ruche](#) ()  
Constructeur par défaut de la classe [Ruche](#).
- [Ruche](#) (Handler [handler](#))
- [Ruche](#) (String Nom)
- [Ruche](#) (String Nom, Handler [handler](#))
- String [getNom](#) ()

*Retourne le nom de la ruche.*

- double `getTemperatureInt` ()
- void `setTemperatureInt` (double `temperatureInt`)
- double `getPoids` ()
- void `setHorodatage` (String `horodatage`)
- void `setPoids` (double `poids`)
- double `getHumiditeInt` ()
- void `setHumiditeInt` (double `humiditeInt`)
- double `getTemperatureExt` ()
- void `setTemperatureExt` (double `temperatureExt`)
- double `getHumiditeExt` ()
- void `setHumiditeExt` (double `humiditeExt`)
- double `getPression` ()
- void `setPression` (double `pression`)
- double `getEnsoleillement` ()
- void `setEnsoleillement` (double `ensoleillement`)
- int `getCharge` ()
- void `setCharge` (int `charge`)
- String `getLatitude` ()
- String `getLongitude` ()
- String `getDescription` ()
- String `getDateDeMiseEnService` ()
- String `getDeviceID` ()
- void `setHandler` (Handler `handler`)

*Fixe le gestionnaire de messages du thread UI.*

- void `recuperer` (final String `Nom`)

*Récupère dans la table `Ruche` l'enregistrement correspondant au nom passé en argument.*

- void `recupererListeRuches` ()
- ArrayList< String > `getListeRuches` ()
- void `recupererChoixChAppID` ()
- ArrayList< String > `getListeChoixAppID` ()
- void `recupererIdTTN` (final String `ApplicationID`)
- int `getIdTTNSelectionne` ()
- int `getIdRuche` ()
- void `recupererMoyennes` (final int `idRuche`)
- void `recupererHistoriqueAlertes` ()
- void `recupererMesuresJournalieresRuche` (final int `idRuche`)
- LineGraphSeries< DataPoint > `getmSeriesTemperaturesInt` ()
- LineGraphSeries< DataPoint > `getmSeriesTemperaturesExt` ()
- LineGraphSeries< DataPoint > `getmSeriesHumiditeInt` ()
- LineGraphSeries< DataPoint > `getmSeriesHumiditeExt` ()
- LineGraphSeries< DataPoint > `getmSeriesEnsoleillement` ()
- LineGraphSeries< DataPoint > `getmSeriesPoids` ()
- LineGraphSeries< DataPoint > `getmSeriesPression` ()
- Double `getEnsoleillement_Basse` ()
- Double `getEnsoleillement_Haute` ()
- Double `getEnsoleillement_Moyenne` ()
- Double `getHum_ext_Basse` ()
- Double `getHum_ext_Haute` ()
- Double `getHum_ext_Moyenne` ()
- Double `getHum_int_Basse` ()
- Double `getHum_int_Haute` ()
- Double `getHum_int_Moyenne` ()
- Double `getPoids_Basse` ()
- Double `getPoids_Haute` ()
- Double `getPoids_Moyenne` ()
- Double `getPression_Basse` ()
- Double `getPression_Haute` ()
- Double `getPression_Moyenne` ()
- Double `getTemp_ext_Basse` ()
- Double `getTemp_ext_Haute` ()
- Double `getTemp_ext_Moyenne` ()
- Double `getTemp_int_Basse` ()
- Double `getTemp_int_Haute` ()
- Double `getTemp_int_Moyenne` ()
- String `getHistoriqueAlertes` ()

#### Attributs privés

- int `idRuche`
- String `nom`
- String `description`
- String `deviceID`
- double `temperatureInt`
- double `humiditeInt`
- double `temperatureExt`
- double `humiditeExt`

```

— double pression
— double ensoleillement
— int charge
— String latitude
— String longitude
— String dateDeMiseEnService
— double poids
— String Horodatage
— BaseDeDonnees bdd = null
— Handler handler = null
— ArrayList< String > listeChoixRuches
— ArrayList< String > listeChoixAppID
— int idTTNSelectionne
— LineGraphSeries< DataPoint > mSeriesTemperaturesInt
— LineGraphSeries< DataPoint > mSeriesTemperaturesExt
— LineGraphSeries< DataPoint > mSeriesHumiditeInt
— LineGraphSeries< DataPoint > mSeriesHumiditeExt
— LineGraphSeries< DataPoint > mSeriesPoids
— LineGraphSeries< DataPoint > mSeriesPression
— LineGraphSeries< DataPoint > mSeriesEnsoleillement
— Double temp_int_Basse
— Double temp_int_Moyenne
— Double temp_int_Haute
— Double hum_int_Basse
— Double hum_int_Moyenne
— Double hum_int_Haute
— Double poids_Basse
— Double poids_Moyenne
— Double poids_Haute
— Double pression_Basse
— Double pression_Moyenne
— Double pression_Haute
— Double ensoleillement_Basse
— Double ensoleillement_Moyenne
— Double ensoleillement_Haute
— Double temp_ext_Basse
— Double temp_ext_Moyenne
— Double temp_ext_Haute
— Double hum_ext_Basse
— Double hum_ext_Moyenne
— Double hum_ext_Haute
— String historiqueAlertes

```

#### Attributs privés statiques

```

— static final String TAG = "Ruche"

```

### 9.35.1 Documentation des constructeurs et destructeur

#### 9.35.1.1 Ruche() [1/4]

```
Ruche.Ruche ( )
```

Constructeur de la classe [Ruche](#).

Constructeur de la classe [Ruche](#) à partir d'un Nom.

#### Paramètres

<i>Nom</i>	String le nom de la ruche dans la table <a href="#">Ruche</a>
------------	---

Définition des pointeurs ensoleillementRuche, humiditeRuche, pressionAtmospheriqueRuche, temperatureRuche, communication↔  
Ruche à 0 (0 = NULL soit aucune adresse)

## Paramètres

<i>donneesRucheTTN</i>	QStringList Informations sur la ruche à créer
<i>parent</i>	QObject Adresse de l'objet Qt parent

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), et [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#).

```

00085         {
00086             Log.d(TAG, "Constructeur par défaut de la classe Ruche");
00087             listeChoixRuches = new ArrayList<String>();
00088             listeChoixAppID = new ArrayList<String>();
00089             if (HoneyBee.BDD) {
00090                 bdd = BaseDeDonnees.getInstance();
00091                 bdd.connecter();
00092             }
00093         }

```

## 9.35.1.2 Ruche() [2/4]

```

fr.campus.laurainc.honeybee.Ruche.Ruche (
    Handler handler )

```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.handler](#).

```

00099         {
00100             Log.d(TAG, "Constructeur par défaut de la classe Ruche");
00101             this.handler = handler;
00102             listeChoixRuches = new ArrayList<String>();
00103             listeChoixAppID = new ArrayList<String>();
00104             mSeriesTemperaturesInt = new LineGraphSeries<>();
00105             mSeriesTemperaturesExt = new LineGraphSeries<>();
00106             mSeriesHumiditeInt = new LineGraphSeries<>();
00107             mSeriesHumiditeExt = new LineGraphSeries<>();
00108             mSeriesPoids = new LineGraphSeries<>();
00109             mSeriesPression = new LineGraphSeries<>();
00110             mSeriesEnsoleillement = new LineGraphSeries<>();
00111             historiqueAlertes = new String();
00112             if (HoneyBee.BDD) {
00113                 bdd = BaseDeDonnees.getInstance();
00114                 bdd.connecter();
00115             }
00116         }

```

## 9.35.1.3 Ruche() [3/4]

```

fr.campus.laurainc.honeybee.Ruche.Ruche (
    String Nom )

```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.recuperer\(\)](#).

```

00123         {
00124             Log.d(TAG, "Constructeur de la classe Ruche -> nom : " + Nom);
00125             this.nom = Nom;
00126             listeChoixRuches = new ArrayList<String>();
00127             listeChoixAppID = new ArrayList<String>();
00128             mSeriesTemperaturesInt = new LineGraphSeries<>();
00129             mSeriesTemperaturesExt = new LineGraphSeries<>();
00130             mSeriesHumiditeInt = new LineGraphSeries<>();
00131             mSeriesHumiditeExt = new LineGraphSeries<>();
00132             mSeriesPoids = new LineGraphSeries<>();
00133             mSeriesPression = new LineGraphSeries<>();
00134             mSeriesEnsoleillement = new LineGraphSeries<>();
00135             if (HoneyBee.BDD) {
00136                 bdd = BaseDeDonnees.getInstance();
00137                 bdd.connecter();
00138                 // Récupères les informations de cette ruche dans la base de données
00139                 recuperer(Nom);
00140             }
00141         }

```

### 9.35.1.4 Ruche() [4/4]

```
fr.campus.laurainc.honeybee.Ruche.Ruche (
    String Nom,
    Handler handler )
```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connecter\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.getInstance\(\)](#), [fr.campus.laurainc.honeybee.Ruche.handler](#), et [fr.campus.laurainc.honeybee.Ruche.recuperer\(\)](#).

```
00148                                     {
00149     Log.d(TAG, "Constructeur de la classe Ruche -> nom : " + Nom);
00150     this.nom = Nom;
00151     listeChoixRuches = new ArrayList<String>();
00152     listeChoixAppID = new ArrayList<String>();
00153     mSeriesTemperaturesInt = new LineGraphSeries<>();
00154     mSeriesTemperaturesExt = new LineGraphSeries<>();
00155     mSeriesHumiditeInt = new LineGraphSeries<>();
00156     mSeriesHumiditeExt = new LineGraphSeries<>();
00157     mSeriesPoids = new LineGraphSeries<>();
00158     mSeriesPression = new LineGraphSeries<>();
00159     mSeriesEnsoleillement = new LineGraphSeries<>();
00160
00161     this.handler = handler;
00162     if (HoneyBee.BDD) {
00163         bdd = BaseDeDonnees.getInstance();
00164         bdd.connecter();
00165         // Récupères les informations de cette ruche dans la base de données
00166         recuperer(nom);
00167     }
00168 }
```

## 9.35.2 Documentation des fonctions membres

### 9.35.2.1 getCharge()

```
int fr.campus.laurainc.honeybee.Ruche.getCharge ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.charge](#).

Référéncé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#).

```
00233                                     {
00234     return charge;
00235 }
```

### 9.35.2.2 getDateDeMiseEnService()

```
String fr.campus.laurainc.honeybee.Ruche.getDateDeMiseEnService ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.dateDeMiseEnService](#).

```
00249                                     {
00250     return dateDeMiseEnService;
00251 }
```



### 9.35.2.3 getDescription()

```
String fr.campus.laurainc.honeybee.Ruche.getDescription ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.description](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#).

```
00245                                     {  
00246         return description;  
00247     }
```

### 9.35.2.4 getDeviceID()

```
String fr.campus.laurainc.honeybee.Ruche.getDeviceID ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.deviceID](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#).

```
00253                                     {  
00254         return deviceID;  
00255     }
```

### 9.35.2.5 getEnseignement()

```
double fr.campus.laurainc.honeybee.Ruche.getEnseignement ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.enseignement](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00227 { return enseignement; }
```

### 9.35.2.6 getEnseignement\_Basse()

```
Double fr.campus.laurainc.honeybee.Ruche.getEnseignement_Basse ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.enseignement\\_Basse](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00939                                     {  
00940         return enseignement_Basse;  
00941     }
```

### 9.35.2.7 getEnseillement\_Haute()

Double fr.campus.laurainc.honeybee.Ruche.getEnseillement\_Haute ( )

Références [fr.campus.laurainc.honeybee.Ruche.enseillement\\_Haute](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
00943                                     {
00944     return enseillement_Haute;
00945 }
```

### 9.35.2.8 getEnseillement\_Moyenne()

Double fr.campus.laurainc.honeybee.Ruche.getEnseillement\_Moyenne ( )

Références [fr.campus.laurainc.honeybee.Ruche.enseillement\\_Moyenne](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
00947                                     {
00948     return enseillement_Moyenne;
00949 }
```

### 9.35.2.9 getHistoriqueAlertes()

String fr.campus.laurainc.honeybee.Ruche.getHistoriqueAlertes ( )

Références [fr.campus.laurainc.honeybee.Ruche.historiqueAlertes](#).

```
01023                                     {
01024     return historiqueAlertes;
01025 }
```

### 9.35.2.10 getHum\_ext\_Basse()

Double fr.campus.laurainc.honeybee.Ruche.getHum\_ext\_Basse ( )

Références [fr.campus.laurainc.honeybee.Ruche.hum\\_ext\\_Basse](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
00951                                     {
00952     return hum_ext_Basse;
00953 }
```

#### 9.35.2.11 getHum\_ext\_Haute()

Double fr.campus.laurainc.honeybee.Ruche.getHum\_ext\_Haute ( )

Références [fr.campus.laurainc.honeybee.Ruche.hum\\_ext\\_Haute](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00955                                     {
00956         return hum_ext_Haute;
00957     }
```

#### 9.35.2.12 getHum\_ext\_Moyenne()

Double fr.campus.laurainc.honeybee.Ruche.getHum\_ext\_Moyenne ( )

Références [fr.campus.laurainc.honeybee.Ruche.hum\\_ext\\_Moyenne](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00959                                     {
00960         return hum_ext_Moyenne;
00961     }
```

#### 9.35.2.13 getHum\_int\_Basse()

Double fr.campus.laurainc.honeybee.Ruche.getHum\_int\_Basse ( )

Références [fr.campus.laurainc.honeybee.Ruche.hum\\_int\\_Basse](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00963                                     {
00964         return hum_int_Basse;
00965     }
```

#### 9.35.2.14 getHum\_int\_Haute()

Double fr.campus.laurainc.honeybee.Ruche.getHum\_int\_Haute ( )

Références [fr.campus.laurainc.honeybee.Ruche.hum\\_int\\_Haute](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00967                                     {
00968         return hum_int_Haute;
00969     }
```

### 9.35.2.15 getHum\_int\_Moyenne()

Double fr.campus.laurainc.honeybee.Ruche.getHum\_int\_Moyenne ( )

Références [fr.campus.laurainc.honeybee.Ruche.hum\\_int\\_Moyenne](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00971                                     {
00972         return hum_int_Moyenne;
00973     }
```

### 9.35.2.16 getHumiditeExt()

double fr.campus.laurainc.honeybee.Ruche.getHumiditeExt ( )

Références [fr.campus.laurainc.honeybee.Ruche.humiditeExt](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00215 { return humiditeExt; }
```

### 9.35.2.17 getHumiditeInt()

double fr.campus.laurainc.honeybee.Ruche.getHumiditeInt ( )

Références [fr.campus.laurainc.honeybee.Ruche.humiditeInt](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00201                                     {
00202         return humiditeInt;
00203     }
```

### 9.35.2.18 getIdRuche()

int fr.campus.laurainc.honeybee.Ruche.getIdRuche ( )

Références [fr.campus.laurainc.honeybee.Ruche.idRuche](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.supprimerRuche\(\)](#).

```
00581                                     {
00582         return idRuche;
00583     }
```

### 9.35.2.19 getIdTTNSelectionne()

```
int fr.campus.laurainc.honeybee.Ruche.getIdTTNSelectionne ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.idTTNSelectionne](#).

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.getIdTTN\(\)](#).

```
00577         {
00578             return idTTNSelectionne;
00579         }
```

### 9.35.2.20 getLatitude()

```
String fr.campus.laurainc.honeybee.Ruche.getLatitude ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.latitude](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.recupererElementsVue\(\)](#).

```
00241 { return latitude;}
```

### 9.35.2.21 getListeChoixAppID()

```
ArrayList<String> fr.campus.laurainc.honeybee.Ruche.getListeChoixAppID ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.listeChoixAppID](#).

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.initialiserChoixAppID\(\)](#).

```
00519     {
00520         return listeChoixAppID;
00521     }
```

### 9.35.2.22 getListeRuches()

```
ArrayList<String> fr.campus.laurainc.honeybee.Ruche.getListeRuches ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.listeChoixRuches](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherListeRuches\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.afficherListeRuches\(\)](#).

```
00459         {
00460             return listeChoixRuches;
00461         }
```

### 9.35.2.23 getLongitude()

`String fr.campus.laurainc.honeybee.Ruche.getLongitude ( )`

Références [fr.campus.laurainc.honeybee.Ruche.longitude](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.DashboardActivity.recupererElementsVue\(\)](#).

```
00243 { return longitude; }
```

### 9.35.2.24 getmSeriesEnsoleillement()

`LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.getmSeriesEnsoleillement ( )`

Références [fr.campus.laurainc.honeybee.Ruche.mSeriesEnsoleillement](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialisationElementsIHM\(\)](#).

```
00927                                     {
00928     return mSeriesEnsoleillement;
00929 }
```

### 9.35.2.25 getmSeriesHumiditeExt()

`LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.getmSeriesHumiditeExt ( )`

Références [fr.campus.laurainc.honeybee.Ruche.mSeriesHumiditeExt](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialisationElementsIHM\(\)](#).

```
00923                                     {
00924     return mSeriesHumiditeExt;
00925 }
```

### 9.35.2.26 getmSeriesHumiditeInt()

`LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.getmSeriesHumiditeInt ( )`

Références [fr.campus.laurainc.honeybee.Ruche.mSeriesHumiditeInt](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialisationElementsIHM\(\)](#).

```
00919                                     {
00920     return mSeriesHumiditeInt;
00921 }
```

### 9.35.2.27 getmSeriesPoids()

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.getmSeriesPoids ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.mSeriesPoids](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialisationElementsIHM\(\)](#).

```
00931                                     {
00932         return mSeriesPoids;
00933     }
```

### 9.35.2.28 getmSeriesPression()

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.getmSeriesPression ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.mSeriesPression](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialisationElementsIHM\(\)](#).

```
00935                                     {
00936         return mSeriesPression;
00937     }
```

### 9.35.2.29 getmSeriesTemperaturesExt()

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.getmSeriesTemperaturesExt ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.mSeriesTemperaturesExt](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialisationElementsIHM\(\)](#).

```
00915                                     {
00916         return mSeriesTemperaturesExt;
00917     }
```

### 9.35.2.30 getmSeriesTemperaturesInt()

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.getmSeriesTemperaturesInt ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.mSeriesTemperaturesInt](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialisationElementsIHM\(\)](#).

```
00910                                     {
00911         Log.d("Graph Données", valueOf(mSeriesTemperaturesInt));
00912         return mSeriesTemperaturesInt;
00913     }
```

### 9.35.2.31 getNom()

```
Ruche.getNom ( )
```

#### Renvoie

String le nom de la ruche dans la table [Ruche](#)

Références [fr.campus.laurainc.honeybee.Ruche.nom](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#).

```
00177         {  
00178         return nom;  
00179     }
```

### 9.35.2.32 getPoids()

```
double fr.campus.laurainc.honeybee.Ruche.getPoids ( )
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
00189         {  
00190         return poids / 1000;  
00191     }
```

### 9.35.2.33 getPoids\_Basse()

```
Double fr.campus.laurainc.honeybee.Ruche.getPoids_Basse ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.poids\\_Basse](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
00975         {  
00976         return poids_Basse;  
00977     }
```

### 9.35.2.34 getPoids\_Haute()

```
Double fr.campus.laurainc.honeybee.Ruche.getPoids_Haute ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.poids\\_Haute](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
00979         {  
00980         return poids_Haute;  
00981     }
```



### 9.35.2.35 getPoids\_Moyenne()

```
Double fr.campus.laurainc.honeybee.Ruche.getPoids_Moyenne ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.poids\\_Moyenne](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00983                                     {
00984     return poids_Moyenne;
00985 }
```

### 9.35.2.36 getPression()

```
double fr.campus.laurainc.honeybee.Ruche.getPression ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.pression](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00221 { return pression; }
```

### 9.35.2.37 getPression\_Basse()

```
Double fr.campus.laurainc.honeybee.Ruche.getPression_Basse ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.pression\\_Basse](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00987                                     {
00988     return pression_Basse;
00989 }
```

### 9.35.2.38 getPression\_Haute()

```
Double fr.campus.laurainc.honeybee.Ruche.getPression_Haute ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.pression\\_Haute](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00991                                     {
00992     return pression_Haute;
00993 }
```

### 9.35.2.39 getPression\_Moyenne()

Double fr.campus.laurainc.honeybee.Ruche.getPression\_Moyenne ( )

Références [fr.campus.laurainc.honeybee.Ruche.pression\\_Moyenne](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00995                                     {
00996         return pression_Moyenne;
00997     }
```

### 9.35.2.40 getTemp\_ext\_Basse()

Double fr.campus.laurainc.honeybee.Ruche.getTemp\_ext\_Basse ( )

Références [fr.campus.laurainc.honeybee.Ruche.temp\\_ext\\_Basse](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
00999                                     {
01000         return temp_ext_Basse;
01001     }
```

### 9.35.2.41 getTemp\_ext\_Haute()

Double fr.campus.laurainc.honeybee.Ruche.getTemp\_ext\_Haute ( )

Références [fr.campus.laurainc.honeybee.Ruche.temp\\_ext\\_Haute](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
01003                                     {
01004         return temp_ext_Haute;
01005     }
```

### 9.35.2.42 getTemp\_ext\_Moyenne()

Double fr.campus.laurainc.honeybee.Ruche.getTemp\_ext\_Moyenne ( )

Références [fr.campus.laurainc.honeybee.Ruche.temp\\_ext\\_Moyenne](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallBacks\(\)](#).

```
01007                                     {
01008         return temp_ext_Moyenne;
01009     }
```

#### 9.35.2.43 getTemp\_int\_Basse()

Double fr.campus.laurainc.honeybee.Ruche.getTemp\_int\_Basse ( )

Références [fr.campus.laurainc.honeybee.Ruche.temp\\_int\\_Basse](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
01011                                     {
01012         return temp_int_Basse;
01013     }
```

#### 9.35.2.44 getTemp\_int\_Haute()

Double fr.campus.laurainc.honeybee.Ruche.getTemp\_int\_Haute ( )

Références [fr.campus.laurainc.honeybee.Ruche.temp\\_int\\_Haute](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
01015                                     {
01016         return temp_int_Haute;
01017     }
```

#### 9.35.2.45 getTemp\_int\_Moyenne()

Double fr.campus.laurainc.honeybee.Ruche.getTemp\_int\_Moyenne ( )

Références [fr.campus.laurainc.honeybee.Ruche.temp\\_int\\_Moyenne](#).

Référencé par [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
01019                                     {
01020         return temp_int_Moyenne;
01021     }
```

#### 9.35.2.46 getTemperatureExt()

double fr.campus.laurainc.honeybee.Ruche.getTemperatureExt ( )

Références [fr.campus.laurainc.honeybee.Ruche.temperatureExt](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
00209 { return temperatureExt; }
```

## 9.35.2.47 getTemperatureInt()

```
double fr.campus.laurainc.honeybee.Ruche.getTemperatureInt ( )
```

Références [fr.campus.laurainc.honeybee.Ruche.temperatureInt](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.afficherInformationsRuche\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.initialiserCallbacks\(\)](#).

```
00181                                     {
00182         return temperatureInt;
00183     }
```

## 9.35.2.48 recuperer()

```
Ruche.recuperer (
    final String Nom )
```

## Renvoi

Nom String le nom de la ruche dans la table [Ruche](#)

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connexion](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.mutex](#), [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_ERREUR](#), et [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_RUCHE](#).

Référencé par [fr.campus.laurainc.honeybee.Ruche.Ruche\(\)](#).

```
00271                                     {
00272         if (!HoneyBee.BDD)
00273             return;
00274
00275         if (bdd.estConnecte()) {
00276             Thread requeteBDD = new Thread(new Runnable() {
00277                 public void run() {
00278                     bdd.mutex.lock();
00279                     try {
00280                         boolean trouve = false;
00281                         String requeteRucheInfo = "SELECT * FROM Ruche WHERE Nom='" + Nom + "'";
00282                         Log.d(TAG, "Requete : " + requeteRucheInfo);
00283                         Statement statement = bdd.connexion.createStatement();
00284                         ResultSet resultSet = statement.executeQuery(requeteRucheInfo);
00285                         if (resultSet.first()) {
00286                             idRuche = resultSet.getInt("idRuche");
00287                             nom = resultSet.getString("Nom");
00288                             description = resultSet.getString("Description");
00289                             deviceId = resultSet.getString("DeviceID");
00290                             description = resultSet.getString("Description");
00291                             longitude = resultSet.getString("Longitude");
00292                             latitude = resultSet.getString("Latitude");
00293                             dateDeMiseEnService = resultSet.getString("DateMiseEnService");
00294                         };
00295                         Log.v(TAG, "BDD -> idRuche : " + idRuche);
00296                         Log.v(TAG, "BDD -> Nom : " + nom);
00297                         Log.v(TAG, "BDD -> Description : " + description);
00298                         Log.d(TAG, "BDD -> DeviceID : " + deviceId);
00299                         Log.v(TAG, "BDD -> Longitude : " + longitude);
00300                         Log.v(TAG, "BDD -> Latitude : " + latitude);
00301                         Log.v(TAG, "BDD -> Date de mise en service : " +
dateDeMiseEnService);
00302                         trouve = true;
00303                     }
00304                     else
00305                         Log.i(TAG, "Aucun résultat obtenu !");
00306
00307                     String requeteMesuresInfoInt = "SELECT * FROM 'MesuresRuche' WHERE 'idRuche'='" +
idRuche + "' ORDER BY 'Horodatage' DESC";
00308                     Log.d(TAG, "Requete : " + requeteMesuresInfoInt);
00309                     ResultSet resultSet2 = statement.executeQuery(requeteMesuresInfoInt);
00310                 }
00311             });
00312         }
```

```

00311         if (resultSet2.first()) {
00312             temperatureInt = resultSet2.getInt("Temperature");
00313             humiditeInt = resultSet2.getInt("Humidite");
00314             Log.v(TAG, "BDD -> Temperature Intérieure: " +
temperatureInt);
00315             Log.v(TAG, "BDD -> Humidite Intérieure: " +
humiditeInt);
00316             trouve = true;
00317         }
00318         else
00319             Log.i(TAG, "Aucun résultat obtenu !");
00320
00321         String requetePoids = "SELECT `Poids` FROM `MesuresPoids` WHERE `idRuche`='" +
idRuche + "' ORDER BY `Horodatage` DESC";
00322         Log.d(TAG, "Requete : " + requetePoids);
00323         ResultSet resultSet3 = statement.executeQuery(requetePoids);
00324
00325         if (resultSet3.first()) {
00326             poids = resultSet3.getInt("Poids");
00327             Log.v(TAG, "BDD -> Poids : " + poids);
00328             trouve = true;
00329         }
00330         else
00331             Log.i(TAG, "Aucun résultat obtenu !");
00332
00333         String requeteMesuresExt = "SELECT * FROM `MesuresEnvironnement` WHERE `idRuche`='" +
+ idRuche + "' ORDER BY `Horodatage` DESC";
00334         Log.d(TAG, "Requete : " + requeteMesuresExt);
00335         ResultSet resultSet4 = statement.executeQuery(requeteMesuresExt);
00336
00337         if (resultSet4.first()) {
00338             temperatureExt = resultSet4.getInt("Temperature");
00339             humiditeExt = resultSet4.getInt("Humidite");
00340             pression = resultSet4.getInt("Pression");
00341             Log.v(TAG, "BDD -> Temperature Extérieure : " +
temperatureExt);
00342             Log.v(TAG, "BDD -> Humidite Extérieure : " +
humiditeExt);
00343             Log.v(TAG, "BDD -> Pression : " + pression);
00344             trouve = true;
00345         }
00346         else
00347             Log.i(TAG, "Aucun résultat obtenu !");
00348
00349         String requeteEnsoleillement = "SELECT `Ensoleillement` FROM
`MesuresEnsoleillement` WHERE `idRuche`='" + idRuche + "' ORDER BY `Horodatage` DESC";
00350         Log.d(TAG, "Requete : " + requeteEnsoleillement);
00351         ResultSet resultSet5 = statement.executeQuery(requeteEnsoleillement);
00352
00353         if (resultSet5.first()) {
00354             ensoleillement = resultSet5.getInt("Ensoleillement");
00355             Log.v(TAG, "BDD -> Ensoleillement : " +
ensoleillement);
00356             trouve = true;
00357         }
00358         else
00359             Log.i(TAG, "Aucun résultat obtenu !");
00360
00361         String requeteCharge = "SELECT `Charge` FROM `MesuresEnergie` WHERE `idRuche`='" +
idRuche + "' ORDER BY `Horodatage` DESC";
00362         Log.d(TAG, "Requete : " + requeteCharge);
00363         ResultSet resultSet6 = statement.executeQuery(requeteCharge);
00364
00365         if (resultSet6.first()) {
00366             charge = resultSet6.getInt("Charge");
00367             Log.v(TAG, "BDD -> Charge : " + charge);
00368             trouve = true;
00369         }
00370         else
00371             Log.i(TAG, "Aucun résultat obtenu !");
00372
00373         if (handler != null && trouve) {
00374             Message msg = Message.obtain();
00375             msg.what = HoneyBee.REQUETE_SQL_RUCHE;
00376             if (handler.sendMessage(msg)) {
00377                 Log.v(TAG, "BDD -> recuperer : message envoyé");
00378             }
00379             else {
00380                 Log.e(TAG, "BDD -> recuperer : message non envoyé !");
00381             }
00382         }
00383     } catch (Exception e) {
00384         e.printStackTrace();
00385         Log.e(TAG, "BDD -> exception : " + e.toString());
00386         if (handler != null) {
00387             Message msg = Message.obtain();
00388             msg.what = HoneyBee.REQUETE_SQL_ERREUR;
00389             handler.sendMessage(msg);
00390         }
00391     } finally {
00392         bdd.mutex.unlock();

```

```

00393         }
00394     }
00395     });
00396
00397     // Démarrage
00398     requeteBDD.start();
00399 } else {
00400     Log.w(TAG, "Pas de connexion MySQL !");
00401 }
00402 }

```

#### 9.35.2.49 recupererChoixChAppID()

```
void fr.campus.laurainc.honeybee.Ruche.recupererChoixChAppID ( )
```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connexion](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.mutex](#), [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_ERREUR](#), et [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_LISTE\\_RUCHES](#).

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.initialiserMenuOnglets\(\)](#).

```

00464     {
00465         listeChoixAppID.clear();
00466
00467         if (!HoneyBee.BDD)
00468             return;
00469
00470         if (bdd.estConnecte()) {
00471             Thread requeteBDD = new Thread(new Runnable() {
00472                 public void run() {
00473                     bdd.mutex.lock();
00474                     try {
00475                         String requeteListesAppID = "SELECT 'ApplicationID' FROM 'TTN'";
00476                         Log.d(TAG, "Requete : " + requeteListesAppID);
00477                         Statement statement = bdd.connexion.createStatement(
00478                             ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_READ_ONLY);
00479                         ResultSet resultSet3 = statement.executeQuery(requeteListesAppID);
00480
00481                         while (resultSet3.next()) {
00482                             listeChoixAppID.add(resultSet3.getString("ApplicationID"));
00483                             int row = resultSet3.getRow();
00484                             Log.v(TAG, "BDD -> Données contenues dans la ligne " + row);
00485                             Log.v(TAG, "BDD -> listeChoixAppID : " +
00486                                 listeChoixAppID.get(row-1));
00487                         }
00488
00489                         if (handler != null) {
00490                             Message msg = Message.obtain();
00491                             msg.what = HoneyBee.REQUETE_SQL_LISTE_RUCHES;
00492                             if (handler.sendMessage(msg)) {
00493                                 Log.v(TAG, "BDD -> recuperer : message envoyé");
00494                             } else {
00495                                 Log.e(TAG, "BDD -> recuperer : message non envoyé !");
00496                             }
00497                         }
00498                     } catch (Exception e) {
00499                         e.printStackTrace();
00500                         Log.e(TAG, "BDD -> exception : " + e.toString());
00501                         if (handler != null) {
00502                             Message msg = Message.obtain();
00503                             msg.what = HoneyBee.REQUETE_SQL_ERREUR;
00504                             handler.sendMessage(msg);
00505                         }
00506                     } finally {
00507                         bdd.mutex.unlock();
00508                     }
00509                 }
00510             });
00511
00512             // Démarrage
00513             requeteBDD.start();
00514         } else {
00515             Log.w(TAG, "Pas de connexion MySQL !");
00516         }
00517     }

```

## 9.35.2.50 recupererHistoriqueAlertes()

```
void fr.campus.laurainc.honeybee.Ruche.recupererHistoriqueAlertes ( )
```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connexion](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.mutex](#), [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_ALERTES](#), et [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_ERREUR](#).

Référencé par [fr.campus.laurainc.honeybee.alertesActivity.onCreate\(\)](#).

```
00712     {
00713         if (!HoneyBee.BDD)
00714             return;
00715
00716         if (bdd.estConnecte()) {
00717             Thread requeteBDD = new Thread(new Runnable() {
00718                 public void run() {
00719                     bdd.mutex.lock();
00720                     try {
00721                         boolean trouve=false;
00722                         String requeteHitoriqueAlertes = "SELECT * FROM Alertes WHERE idRuche='6'";
00723                         Statement statement = bdd.connexion.createStatement(ResultSet.
TYPE_FORWARD_ONLY, ResultSet.CONCUR_READ_ONLY);
00724
00725                         ResultSet resultSet = statement.executeQuery(requeteHitoriqueAlertes);
00726                         Log.d(TAG, "Requete : " + requeteHitoriqueAlertes);
00727                         while (resultSet.next()) {
00728                             historiqueAlertes += resultSet.getString("Description") + " "
+ resultSet.getString("Horodatage")+"\n";
00729                             trouve = true;
00730                         }
00731
00732                         Log.d(TAG, "Alertes : " + historiqueAlertes);
00733
00734
00735                         if (handler != null && trouve) {
00736                             Message msg = Message.obtain();
00737                             msg.what = HoneyBee.REQUETE_SQL_ALERTES;
00738                             if (handler.sendMessage(msg)) {
00739                                 Log.v(TAG, "BDD -> recuperer : message envoyé");
00740                             }
00741                             else {
00742                                 Log.e(TAG, "BDD -> recuperer : message non envoyé !");
00743                             }
00744                         }
00745                     } catch (Exception e) {
00746                         e.printStackTrace();
00747                         Log.e(TAG, "BDD -> exception : " + e.toString());
00748                         if (handler != null) {
00749                             Message msg = Message.obtain();
00750                             msg.what = HoneyBee.REQUETE_SQL_ERREUR;
00751                             handler.sendMessage(msg);
00752                         }
00753                     } finally {
00754                         bdd.mutex.unlock();
00755                     }
00756                 }
00757             });
00758
00759             // Démarrage
00760             requeteBDD.start();
00761         } else {
00762             Log.w(TAG, "Pas de connexion MySQL !");
00763         }
00764     }
00765 }
```

## 9.35.2.51 recupererIdTTN()

```
void fr.campus.laurainc.honeybee.Ruche.recupererIdTTN (
    final String ApplicationID )
```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connexion](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.mutex](#), [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_ERREUR](#), et [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_IDTTN](#).

Référencé par [fr.campus.laurainc.honeybee.NouvelleRucheActivity.initialiserChoixAppID\(\)](#).

```

00524      {
00525          if (!HoneyBee.BDD)
00526              return;
00527
00528          if (bdd.estConnecte()) {
00529              Thread requeteBDD = new Thread(new Runnable() {
00530                  public void run() {
00531                      bdd.mutex.lock();
00532                      try {
00533                          boolean trouve = false;
00534                          String requeteIdTTN = "SELECT idTTN FROM TTN WHERE ApplicationID=\"" + ApplicationID+ "
00535                          \";
00536                          Log.d(TAG, "Requete : " + requeteIdTTN);
00537                          Statement statement = bdd.connexion.createStatement();
00538                          ResultSet resultSet = statement.executeQuery(requeteIdTTN);
00539                          if (resultSet.first()) {
00540                              idTTNSelectionne = resultSet.getInt("idTTN");
00541                              Log.v(TAG, "BDD -> idTTN : " + idTTNSelectionne);
00542                              trouve = true;
00543                          }
00544                          else
00545                              Log.i(TAG, "Aucun résultat obtenu !");
00546
00547                          if (handler != null && trouve) {
00548                              Message msg = Message.obtain();
00549                              msg.what = HoneyBee.REQUETE_SQL_IDTTN;
00550                              if (handler.sendMessage(msg)) {
00551                                  Log.v(TAG, "BDD -> recuperer : message envoyé");
00552                              }
00553                              else {
00554                                  Log.e(TAG, "BDD -> recuperer : message non envoyé !");
00555                              }
00556                          }
00557                      } catch (Exception e) {
00558                          e.printStackTrace();
00559                          Log.e(TAG, "BDD -> exception : " + e.toString());
00560                          if (handler != null) {
00561                              Message msg = Message.obtain();
00562                              msg.what = HoneyBee.REQUETE_SQL_ERREUR;
00563                              handler.sendMessage(msg);
00564                          }
00565                      } finally {
00566                          bdd.mutex.unlock();
00567                      }
00568                  }
00569              });
00570
00571              // Démarrage
00572              requeteBDD.start();
00573          } else {
00574              Log.w(TAG, "Pas de connexion MySQL !");
00575          }
00576      }

```

### 9.35.2.52 recupererListeRuches()

```
void fr.campus.laurainc.honeybee.Ruche.recupererListeRuches ( )
```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connexion](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.mutex](#), [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_ERREUR](#), et [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_LISTE\\_RUCHES](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.onCreate\(\)](#), et [fr.campus.laurainc.honeybee.GraphActivity.onCreate\(\)](#).

```

00404      {
00405
00406          listeChoixRuches.clear();
00407
00408          if (!HoneyBee.BDD)
00409              return;
00410
00411          if (bdd.estConnecte()) {
00412              Thread requeteBDD = new Thread(new Runnable() {
00413                  public void run() {
00414                      bdd.mutex.lock();
00415                      try {
00416                          String requeteListesRuches = "SELECT Nom FROM `Ruche`";
00417                          Log.d(TAG, "Requete : " + requeteListesRuches);
00418                          Statement statement = bdd.connexion.createStatement(

```



```

00419         ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_READ_ONLY);
00420         ResultSet resultSet3 = statement.executeQuery(requeteListesRuches);
00421
00422         while (resultSet3.next()) {
00423             listeChoixRuches.add(resultSet3.getString("Nom"));
00424             int row = resultSet3.getRow();
00425             Log.v(TAG, "BDD -> Données contenues dans la ligne " + row);
00426             Log.v(TAG, "BDD -> listeRuches : " +
listeChoixRuches.get(row-1));
00427         }
00428
00429         if (handler != null) {
00430             Message msg = Message.obtain();
00431             msg.what = HoneyBee.REQUETE_SQL_LISTE_RUCHES;
00432             if (handler.sendMessage(msg)) {
00433                 Log.v(TAG, "BDD -> recuperer : message envoyé");
00434             } else {
00435                 Log.e(TAG, "BDD -> recuperer : message non envoyé !");
00436             }
00437         }
00438     } catch (Exception e) {
00439         e.printStackTrace();
00440         Log.e(TAG, "BDD -> exception : " + e.toString());
00441         if (handler != null) {
00442             Message msg = Message.obtain();
00443             msg.what = HoneyBee.REQUETE_SQL_ERREUR;
00444             handler.sendMessage(msg);
00445         }
00446     } finally {
00447         bdd.mutex.unlock();
00448     }
00449 }
00450 });
00451
00452 // Démarrage
00453 requeteBDD.start();
00454 } else {
00455     Log.w(TAG, "Pas de connexion MySQL !");
00456 }
00457 }

```

### 9.35.2.53 recupererMesuresJournalieresRuche()

```

void fr.campus.laurainc.honeybee.Ruche.recupererMesuresJournalieresRuche (
    final int idRuche )

```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connexion](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.mutex](#), [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_ERREUR](#), et [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_MESURES\\_RUCHES](#).

```

00767
00768
00769     if (!HoneyBee.BDD)
00770         return;
00771
00772     if (bdd.estConnecte()) {
00773         Thread requeteBDD = new Thread(new Runnable() {
00774             public void run() {
00775                 bdd.mutex.lock();
00776                 try {
00777                     Calendar calendar = Calendar.getInstance();
00778                     SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd");
00779                     final String aujourd'hui = simpleDateFormat.format(calendar.getTime());
00780                     String requeteMesuresJournalieresRuche = "SELECT * FROM MesuresJournalieresRuche
WHERE DateMesure='" + aujourd'hui + "' AND idRuche='" + idRuche + "' ORDER BY HeureMesure ASC";
00781                     String requeteMesuresJournalieresPoids = "SELECT * FROM MesuresJournalieresPoids
WHERE DateMesure='" + aujourd'hui + "' AND idRuche='" + idRuche + "' ORDER BY HeureMesure ASC";
00782                     String requeteMesuresJournalieresEnvironnement = "SELECT * FROM
MesuresJournalieresEnvironnement WHERE DateMesure='" + aujourd'hui + "' AND idRuche='" + idRuche + "' ORDER BY HeureMesure ASC";
00783                     String requeteMesuresJournalieresEnsoleillement = "SELECT * FROM
MesuresJournalieresEnsoleillement WHERE DateMesure='" + aujourd'hui + "' AND idRuche='" + idRuche + "' ORDER BY HeureMesure ASC";
00784                     Statement statement = bdd.connexion.createStatement(ResultSet.
TYPE_FORWARD_ONLY, ResultSet.CONCUR_READ_ONLY);
00785
00786
00787                     ResultSet resultSet = statement.executeQuery(requeteMesuresJournalieresRuche);
00788

```

```

00789         while (resultSet.next()) {
00790             Log.d(TAG, "Requete : " + requeteMesuresJournalieresRuche);
00791             SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
00792             Date heureMesure = sdf.parse(resultSet.getString("HeureMesure"));
00793             calendar.setTime(heureMesure);
00794             int heure = calendar.get(Calendar.HOUR_OF_DAY);
00795
00796             Log.v(TAG, "BDD -> Temperature " + resultSet.getDouble("Temperature"));
00797             Log.v(TAG, "BDD -> HeureMesure : " + resultSet.getString("HeureMesure"));
00798             Log.v(TAG, "BDD -> Heure : " + heure);
00799             mSeriesTemperaturesInt.appendData(new DataPoint(heure,
resultSet.getDouble("Temperature")), true, 24);
00800             temp_int_Basse = resultSet.getDouble("TemperatureMin");
00801             temp_int_Haute = resultSet.getDouble("TemperatureMax");
00802
00803             Log.v(TAG, "BDD -> Humidite " + resultSet.getDouble("Humidite"));
00804             Log.v(TAG, "BDD -> HeureMesure : " + resultSet.getString("HeureMesure"));
00805             Log.v(TAG, "BDD -> Heure : " + heure);
00806             mSeriesHumiditeInt.appendData(new DataPoint(heure, resultSet.
getDouble("Humidite")), true, 24);
00807             hum_int_Basse = resultSet.getDouble("HumiditeMin");
00808             hum_int_Haute = resultSet.getDouble("HumiditeMax");
00809         }
00810
00811
00812         ResultSet resultSet2 = statement.executeQuery(requeteMesuresJournalieresPoids);
00813
00814         while (resultSet2.next()) {
00815             Log.d(TAG, "Requete : " + requeteMesuresJournalieresPoids);
00816             SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
00817             Date heureMesure = sdf.parse(resultSet2.getString("HeureMesure"));
00818             calendar.setTime(heureMesure);
00819             int heure = calendar.get(Calendar.HOUR_OF_DAY);
00820
00821             Log.v(TAG, "BDD -> Poids " + resultSet2.getDouble("Poids"));
00822             Log.v(TAG, "BDD -> HeureMesure : " + resultSet2.getString("HeureMesure"));
00823             Log.v(TAG, "BDD -> Heure : " + heure);
00824             mSeriesPoids.appendData(new DataPoint(heure, resultSet2.getDouble("
Poids")), true, 24);
00825             poids_Basse= resultSet2.getDouble("PoidsMin");
00826             poids_Haute = resultSet2.getDouble("PoidsMax");
00827         }
00828
00829
00830         ResultSet resultSet3 = statement.executeQuery(
requeteMesuresJournalieresEnvironnement);
00831
00832         while (resultSet3.next()) {
00833             Log.d(TAG, "Requete : " + requeteMesuresJournalieresEnvironnement);
00834             SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
00835             Date heureMesure = sdf.parse(resultSet3.getString("HeureMesure"));
00836             calendar.setTime(heureMesure);
00837             int heure = calendar.get(Calendar.HOUR_OF_DAY);
00838
00839             Log.v(TAG, "BDD -> Temperature " + resultSet3.getDouble("Temperature"));
00840             Log.v(TAG, "BDD -> HeureMesure : " + resultSet3.getString("HeureMesure"));
00841             Log.v(TAG, "BDD -> Heure : " + heure);
00842             mSeriesTemperaturesExt.appendData(new DataPoint(heure,
resultSet3.getDouble("Temperature")), true, 24);
00843             temp_ext_Basse = resultSet3.getDouble("TemperatureMin");
00844             temp_ext_Haute = resultSet3.getDouble("TemperatureMax");
00845
00846             Log.v(TAG, "BDD -> Humidite " + resultSet3.getDouble("Humidite"));
00847             Log.v(TAG, "BDD -> HeureMesure : " + resultSet3.getString("HeureMesure"));
00848             Log.v(TAG, "BDD -> Heure : " + heure);
00849             mSeriesHumiditeExt.appendData(new DataPoint(heure, resultSet3
.getDouble("Humidite")), true, 24);
00850             hum_ext_Basse = resultSet3.getDouble("HumiditeMin");
00851             hum_ext_Haute = resultSet3.getDouble("HumiditeMax");
00852
00853             Log.v(TAG, "BDD -> Pression " + resultSet3.getDouble("Pression"));
00854             Log.v(TAG, "BDD -> HeureMesure : " + resultSet3.getString("HeureMesure"));
00855             Log.v(TAG, "BDD -> Heure : " + heure);
00856             mSeriesPression.appendData(new DataPoint(heure, resultSet3.
getDouble("Pression")), true, 24);
00857             pression_Basse = resultSet3.getDouble("PressionMin");
00858             pression_Haute = resultSet3.getDouble("PressionMax");
00859         }
00860
00861
00862         ResultSet resultSet4 = statement.executeQuery(
requeteMesuresJournalieresEnsoleillement);
00863
00864         while (resultSet4.next()) {
00865             Log.d(TAG, "Requete : " + requeteMesuresJournalieresEnsoleillement);
00866             SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
00867             Date heureMesure = sdf.parse(resultSet4.getString("HeureMesure"));
00868             calendar.setTime(heureMesure);
00869             int heure = calendar.get(Calendar.HOUR_OF_DAY);
00870
00871             Log.v(TAG, "BDD -> Ensoleillement " + resultSet4.getDouble("Ensoleillement")

```

```

);
00872         Log.v(TAG, "BDD -> HeureMesure : " + resultSet4.getString("HeureMesure"));
00873         Log.v(TAG, "BDD -> Heure : " + heure);
00874         mSeriesEnsoleillement.appendData(new DataPoint(heure,
resultSet4.getDouble("Ensoleillement")), true, 24);
00875         ensoleillement_Basse = resultSet4.getDouble("
EnsoleillementMin");
00876         ensoleillement_Haute = resultSet4.getDouble("
EnsoleillementMax");
00877     }
00878
00879
00880     if (handler != null) {
00881         Message msg = Message.obtain();
00882         msg.what = HoneyBee.REQUETE_SQL_MESURES_RUCHES;
00883         if (handler.sendMessage(msg)) {
00884             Log.v(TAG, "BDD -> recuperer : message envoyé");
00885         } else {
00886             Log.e(TAG, "BDD -> recuperer : message non envoyé !");
00887         }
00888     }
00889 } catch (Exception e) {
00890     e.printStackTrace();
00891     Log.e(TAG, "BDD -> exception : " + e.toString());
00892     if (handler != null) {
00893         Message msg = Message.obtain();
00894         msg.what = HoneyBee.REQUETE_SQL_ERREUR;
00895         handler.sendMessage(msg);
00896     }
00897 } finally {
00898     bdd.mutex.unlock();
00899 }
00900 }
00901 });
00902
00903 // Démarrage
00904 requeteBDD.start();
00905 } else {
00906     Log.w(TAG, "Pas de connexion MySQL !");
00907 }
00908 }

```

### 9.35.2.54 recupererMoyennes()

```

void fr.campus.laurainc.honeybee.Ruche.recupererMoyennes (
    final int idRuche )

```

Références [fr.campus.laurainc.honeybee.HoneyBee.BDD](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.connexion](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.estConnecte\(\)](#), [fr.campus.laurainc.honeybee.BaseDeDonnees.mutex](#), [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_ERREUR](#), et [fr.campus.laurainc.honeybee.HoneyBee.REQUETE\\_SQL\\_IDTTN](#).

```

00586     {
00587         if (!HoneyBee.BDD)
00588             return;
00589
00590         if (bdd.estConnecte()) {
00591             Thread requeteBDD = new Thread(new Runnable() {
00592                 public void run() {
00593                     bdd.mutex.lock();
00594                     try {
00595                         boolean trouve=false;
00596                         String requeteMoyenneTempInt = "SELECT AVG(Temperature) FROM MesuresRuche WHERE
idRuche='" + idRuche + "'";
00597                         String requeteMoyenneTempExt = "SELECT AVG(Temperature) FROM MesuresEnvironnement
WHERE idRuche='" + idRuche + "'";
00598                         String requeteMoyenneHumInt = "SELECT AVG(Humidite) FROM MesuresRuche WHERE
idRuche='" + idRuche + "'";
00599                         String requeteMoyenneHumExt = "SELECT AVG(Humidite) FROM MesuresEnvironnement WHERE
idRuche='" + idRuche + "'";
00600                         String requeteMoyennePression = "SELECT AVG(Pression) FROM MesuresEnvironnement
WHERE idRuche='" + idRuche + "'";
00601                         String requeteMoyenneEnsoleillement = "SELECT AVG(Ensoleillement) FROM
MesuresEnsoleillement WHERE idRuche='" + idRuche + "'";
00602                         String requeteMoyennePoids = "SELECT AVG(Poids) FROM MesuresPoids WHERE idRuche='"
+ idRuche + "'";
00603                         Statement statement = bdd.connexion.createStatement();
00604                         ResultSet resultSet = statement.executeQuery(requeteMoyenneTempInt);
00605                         if (resultSet.first())
00606                             {
00607

```

```

00608         Log.d(TAG, "Requete :" + requeteMoyenneTempInt);
00609         temp_int_Moyenne = resultSet.getDouble("AVG(Temperature)");
00610         trouve = true;
00611     }
00612     else
00613         Log.i(TAG, "Aucun résultat obtenu !");
00614
00615     ResultSet resultSet2 = statement.executeQuery(requeteMoyenneTempExt);
00616     if (resultSet2.first())
00617     {
00618         Log.d(TAG, "Requete :" + requeteMoyenneTempExt);
00619         temp_ext_Moyenne = resultSet2.getDouble("AVG(Temperature)");
00620         trouve = true;
00621     }
00622     else
00623         Log.i(TAG, "Aucun résultat obtenu !");
00624
00625     ResultSet resultSet3 = statement.executeQuery(requeteMoyenneHumInt);
00626     if (resultSet3.first())
00627     {
00628         Log.d(TAG, "Requete :" + requeteMoyenneHumInt);
00629         hum_int_Moyenne = resultSet3.getDouble("AVG(Humidite)");
00630         trouve = true;
00631     }
00632     else
00633         Log.i(TAG, "Aucun résultat obtenu !");
00634
00635     ResultSet resultSet4 = statement.executeQuery(requeteMoyenneHumExt);
00636     if (resultSet4.first())
00637     {
00638         Log.d(TAG, "Requete :" + requeteMoyenneHumExt);
00639         hum_ext_Moyenne = resultSet4.getDouble("AVG(Humidite)");
00640         trouve = true;
00641     }
00642     else
00643         Log.i(TAG, "Aucun résultat obtenu !");
00644
00645     ResultSet resultSet5 = statement.executeQuery(requeteMoyenneEnsoleillement);
00646     if (resultSet5.first())
00647     {
00648         Log.d(TAG, "Requete :" + requeteMoyenneEnsoleillement);
00649         ensoleillement_Moyenne = resultSet5.getDouble("
AVG(Ensoleillement)");
00650         trouve = true;
00651     }
00652     else
00653         Log.i(TAG, "Aucun résultat obtenu !");
00654
00655     ResultSet resultSet6 = statement.executeQuery(requeteMoyennePoids);
00656     if (resultSet6.first())
00657     {
00658         Log.d(TAG, "Requete :" + requeteMoyennePoids);
00659         poids_Moyenne = resultSet6.getDouble("AVG(Poids)");
00660         trouve = true;
00661     }
00662     else
00663         Log.i(TAG, "Aucun résultat obtenu !");
00664
00665
00666     ResultSet resultSet7 = statement.executeQuery(requeteMoyennePression);
00667     if (resultSet7.first())
00668     {
00669         Log.d(TAG, "Requete :" + requeteMoyennePression);
00670         pression_Moyenne = resultSet7.getDouble("AVG(Pression)");
00671         trouve = true;
00672     }
00673     else
00674         Log.i(TAG, "Aucun résultat obtenu !");
00675
00676
00677
00678
00679     if (handler != null && trouve) {
00680         Message msg = Message.obtain();
00681         msg.what = HoneyBee.REQUETE_SQL_IDTTN;
00682         if (handler.sendMessage(msg)) {
00683             Log.v(TAG, "BDD -> recuperer : message envoyé");
00684         }
00685         else {
00686             Log.e(TAG, "BDD -> recuperer : message non envoyé !");
00687         }
00688     }
00689 } catch (Exception e) {
00690     e.printStackTrace();
00691     Log.e(TAG, "BDD -> exception : " + e.toString());
00692     if (handler != null) {
00693         Message msg = Message.obtain();
00694         msg.what = HoneyBee.REQUETE_SQL_ERREUR;
00695         handler.sendMessage(msg);
00696     }
00697 } finally {

```

```

00698             bdd.mutex.unlock();
00699         }
00700     }
00701 });
00702
00703     // Démarrage
00704     requeteBDD.start();
00705 } else {
00706     Log.w(TAG, "Pas de connexion MySQL !");
00707 }
00708
00709 }

```

#### 9.35.2.55 setCharge()

```

void fr.campus.laurainc.honeybee.Ruche.setCharge (
    int charge )

```

Références [fr.campus.laurainc.honeybee.Ruche.charge](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```

00237         {
00238             this.charge = charge;
00239         }

```

#### 9.35.2.56 setEnseillement()

```

void fr.campus.laurainc.honeybee.Ruche.setEnseillement (
    double enseillement )

```

Références [fr.campus.laurainc.honeybee.Ruche.enseillement](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```

00229         {
00230             this.enseillement = enseillement;
00231         }

```

#### 9.35.2.57 setHandler()

```

Ruche.setHandler (
    Handler handler )

```

##### Paramètres

<i>handler</i>	Handler le gestionnaire de messages du thread UI
----------------	--

Références [fr.campus.laurainc.honeybee.Ruche.handler](#).

```

00262         {
00263             this.handler = handler;
00264         }

```

**9.35.2.58 setHorodatage()**

```
void fr.campus.laurainc.honeybee.Ruche.setHorodatage (
    String horodatage )
```

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00193                                     {
00194     Horodatage = horodatage;
00195 }
```

**9.35.2.59 setHumiditeExt()**

```
void fr.campus.laurainc.honeybee.Ruche.setHumiditeExt (
    double humiditeExt )
```

Références [fr.campus.laurainc.honeybee.Ruche.humiditeExt](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00217                                     {
00218     this.humiditeExt = humiditeExt;
00219 }
```

**9.35.2.60 setHumiditeInt()**

```
void fr.campus.laurainc.honeybee.Ruche.setHumiditeInt (
    double humiditeInt )
```

Références [fr.campus.laurainc.honeybee.Ruche.humiditeInt](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00205                                     {
00206     this.humiditeInt = humiditeInt;
00207 }
```

**9.35.2.61 setPoids()**

```
void fr.campus.laurainc.honeybee.Ruche.setPoids (
    double poids )
```

Références [fr.campus.laurainc.honeybee.Ruche.poids](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00197                                     {
00198     this.poids = poids;
00199 }
```

### 9.35.2.62 setPression()

```
void fr.campus.laurainc.honeybee.Ruche.setPression (
    double pression )
```

Références [fr.campus.laurainc.honeybee.Ruche.pression](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00223                                     {
00224         this.pression = pression;
00225     }
```

### 9.35.2.63 setTemperatureExt()

```
void fr.campus.laurainc.honeybee.Ruche.setTemperatureExt (
    double temperatureExt )
```

Références [fr.campus.laurainc.honeybee.Ruche.temperatureExt](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00211                                     {
00212         this.temperatureExt = temperatureExt;
00213     }
```

### 9.35.2.64 setTemperatureInt()

```
void fr.campus.laurainc.honeybee.Ruche.setTemperatureInt (
    double temperatureInt )
```

Références [fr.campus.laurainc.honeybee.Ruche.temperatureInt](#).

Référencé par [fr.campus.laurainc.honeybee.DashboardActivity.traiterMessage\(\)](#).

```
00185                                     {
00186         this.temperatureInt = temperatureInt;
00187     }
```

## 9.35.3 Documentation des données membres

### 9.35.3.1 bdd

```
BaseDeDonnees fr.campus.laurainc.honeybee.Ruche.bdd = null [private]
```

### 9.35.3.2 charge

```
int fr.campus.laurainc.honeybee.Ruche.charge [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getCharge\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.setCharge\(\)](#).

### 9.35.3.3 dateDeMiseEnService

```
String fr.campus.laurainc.honeybee.Ruche.dateDeMiseEnService [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getDateDeMiseEnService\(\)](#).

### 9.35.3.4 description

```
String fr.campus.laurainc.honeybee.Ruche.description [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getDescription\(\)](#).

### 9.35.3.5 deviceID

```
String fr.campus.laurainc.honeybee.Ruche.deviceID [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getDeviceID\(\)](#).

### 9.35.3.6 ensoleillement

```
double fr.campus.laurainc.honeybee.Ruche.ensoleillement [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getEnsoleillement\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.setEnsoleillement\(\)](#).

### 9.35.3.7 ensoleillement\_Basse

```
Double fr.campus.laurainc.honeybee.Ruche.ensoleillement_Basse [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getEnsoleillement\\_Basse\(\)](#).

### 9.35.3.8 ensoleillement\_Haute

```
Double fr.campus.laurainc.honeybee.Ruche.ensoleillement_Haute [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getEnsoleillement\\_Haute\(\)](#).



#### 9.35.3.9 `ensoileillement_Moyenne`

```
Double fr.campus.laurainc.honeybee.Ruche.ensoileillement_Moyenne [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getEnsoileillement\\_Moyenne\(\)](#).

#### 9.35.3.10 `handler`

```
Handler fr.campus.laurainc.honeybee.Ruche.handler = null [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.Ruche\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.setHandler\(\)](#).

#### 9.35.3.11 `historiqueAlertes`

```
String fr.campus.laurainc.honeybee.Ruche.historiqueAlertes [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHistoriqueAlertes\(\)](#).

#### 9.35.3.12 `Horodatage`

```
String fr.campus.laurainc.honeybee.Ruche.Horodatage [private]
```

#### 9.35.3.13 `hum_ext_Basse`

```
Double fr.campus.laurainc.honeybee.Ruche.hum_ext_Basse [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHum\\_ext\\_Basse\(\)](#).

#### 9.35.3.14 `hum_ext_Haute`

```
Double fr.campus.laurainc.honeybee.Ruche.hum_ext_Haute [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHum\\_ext\\_Haute\(\)](#).

#### 9.35.3.15 `hum_ext_Moyenne`

```
Double fr.campus.laurainc.honeybee.Ruche.hum_ext_Moyenne [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHum\\_ext\\_Moyenne\(\)](#).

**9.35.3.16 hum\_int\_Basse**

Double fr.campus.laurainc.honeybee.Ruche.hum\_int\_Basse [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHum\\_int\\_Basse\(\)](#).

**9.35.3.17 hum\_int\_Haute**

Double fr.campus.laurainc.honeybee.Ruche.hum\_int\_Haute [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHum\\_int\\_Haute\(\)](#).

**9.35.3.18 hum\_int\_Moyenne**

Double fr.campus.laurainc.honeybee.Ruche.hum\_int\_Moyenne [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHum\\_int\\_Moyenne\(\)](#).

**9.35.3.19 humiditeExt**

double fr.campus.laurainc.honeybee.Ruche.humiditeExt [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHumiditeExt\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.setHumiditeExt\(\)](#).

**9.35.3.20 humiditeInt**

double fr.campus.laurainc.honeybee.Ruche.humiditeInt [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getHumiditeInt\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.setHumiditeInt\(\)](#).

**9.35.3.21 idRuche**

int fr.campus.laurainc.honeybee.Ruche.idRuche [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getIdRuche\(\)](#).

**9.35.3.22 idTTNSelectionne**

int fr.campus.laurainc.honeybee.Ruche.idTTNSelectionne [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getIdTTNSelectionne\(\)](#).

### 9.35.3.23 latitude

```
String fr.campus.laurainc.honeybee.Ruche.latitude [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getLatitude\(\)](#).

### 9.35.3.24 listeChoixAppID

```
ArrayList<String> fr.campus.laurainc.honeybee.Ruche.listeChoixAppID [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getListeChoixAppID\(\)](#).

### 9.35.3.25 listeChoixRuches

```
ArrayList<String> fr.campus.laurainc.honeybee.Ruche.listeChoixRuches [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getListeRuches\(\)](#).

### 9.35.3.26 longitude

```
String fr.campus.laurainc.honeybee.Ruche.longitude [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getLongitude\(\)](#).

### 9.35.3.27 mSeriesEnsoleillement

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.mSeriesEnsoleillement [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getMSeriesEnsoleillement\(\)](#).

### 9.35.3.28 mSeriesHumiditeExt

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.mSeriesHumiditeExt [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getMSeriesHumiditeExt\(\)](#).

### 9.35.3.29 mSeriesHumiditeInt

```
LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.mSeriesHumiditeInt [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getMSeriesHumiditeInt\(\)](#).

#### 9.35.3.30 mSeriesPoids

`LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.mSeriesPoids [private]`

Référencé par [fr.campus.laurainc.honeybee.Ruche.getmSeriesPoids\(\)](#).

#### 9.35.3.31 mSeriesPression

`LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.mSeriesPression [private]`

Référencé par [fr.campus.laurainc.honeybee.Ruche.getmSeriesPression\(\)](#).

#### 9.35.3.32 mSeriesTemperaturesExt

`LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.mSeriesTemperaturesExt [private]`

Référencé par [fr.campus.laurainc.honeybee.Ruche.getmSeriesTemperaturesExt\(\)](#).

#### 9.35.3.33 mSeriesTemperaturesInt

`LineGraphSeries<DataPoint> fr.campus.laurainc.honeybee.Ruche.mSeriesTemperaturesInt [private]`

Référencé par [fr.campus.laurainc.honeybee.Ruche.getmSeriesTemperaturesInt\(\)](#).

#### 9.35.3.34 nom

`String fr.campus.laurainc.honeybee.Ruche.nom [private]`

Référencé par [fr.campus.laurainc.honeybee.Ruche.getNom\(\)](#).

#### 9.35.3.35 poids

`double fr.campus.laurainc.honeybee.Ruche.poids [private]`

Référencé par [fr.campus.laurainc.honeybee.Ruche.setPoids\(\)](#).

#### 9.35.3.36 poids\_Basse

`Double fr.campus.laurainc.honeybee.Ruche.poids_Basse [private]`

Référencé par [fr.campus.laurainc.honeybee.Ruche.getPoids\\_Basse\(\)](#).

**9.35.3.37 poids\_Haute**

```
Double fr.campus.laurainc.honeybee.Ruche.poids_Haute [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getPoids\\_Haute\(\)](#).

**9.35.3.38 poids\_Moyenne**

```
Double fr.campus.laurainc.honeybee.Ruche.poids_Moyenne [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getPoids\\_Moyenne\(\)](#).

**9.35.3.39 pression**

```
double fr.campus.laurainc.honeybee.Ruche.pression [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getPression\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.setPression\(\)](#).

**9.35.3.40 pression\_Basse**

```
Double fr.campus.laurainc.honeybee.Ruche.pression_Basse [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getPression\\_Basse\(\)](#).

**9.35.3.41 pression\_Haute**

```
Double fr.campus.laurainc.honeybee.Ruche.pression_Haute [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getPression\\_Haute\(\)](#).

**9.35.3.42 pression\_Moyenne**

```
Double fr.campus.laurainc.honeybee.Ruche.pression_Moyenne [private]
```

Référencé par [fr.campus.laurainc.honeybee.Ruche.getPression\\_Moyenne\(\)](#).

**9.35.3.43 TAG**

```
final String fr.campus.laurainc.honeybee.Ruche.TAG = "Ruche" [static], [private]
```

**9.35.3.44 temp\_ext\_Basse**

Double fr.campus.laurainc.honeybee.Ruche.temp\_ext\_Basse [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getTemp\\_ext\\_Basse\(\)](#).

**9.35.3.45 temp\_ext\_Haute**

Double fr.campus.laurainc.honeybee.Ruche.temp\_ext\_Haute [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getTemp\\_ext\\_Haute\(\)](#).

**9.35.3.46 temp\_ext\_Moyenne**

Double fr.campus.laurainc.honeybee.Ruche.temp\_ext\_Moyenne [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getTemp\\_ext\\_Moyenne\(\)](#).

**9.35.3.47 temp\_int\_Basse**

Double fr.campus.laurainc.honeybee.Ruche.temp\_int\_Basse [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getTemp\\_int\\_Basse\(\)](#).

**9.35.3.48 temp\_int\_Haute**

Double fr.campus.laurainc.honeybee.Ruche.temp\_int\_Haute [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getTemp\\_int\\_Haute\(\)](#).

**9.35.3.49 temp\_int\_Moyenne**

Double fr.campus.laurainc.honeybee.Ruche.temp\_int\_Moyenne [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getTemp\\_int\\_Moyenne\(\)](#).

**9.35.3.50 temperatureExt**

double fr.campus.laurainc.honeybee.Ruche.temperatureExt [private]

Référencé par [fr.campus.laurainc.honeybee.Ruche.getTemperatureExt\(\)](#), et [fr.campus.laurainc.honeybee.Ruche.setTemperatureExt\(\)](#).



- void [recevoirMesurePoidsTraite](#) (double poids, QString)
- void [recevoirTraitementTemperatureInterieur](#) (const double temperatureInterieurMoyenne, const double temperatureInterieur←  
Minimum, const double temperatureInterieurMaximum, int heure)
- void [recevoirTraitementTemperatureExterieur](#) (const double temperatureExterieurMoyenne, const double temperature←  
ExterieurMinimum, const double temperatureExterieurMaximum, int heure)
- void [recevoirTraitementHumiditeInterieur](#) (const double temperatureInterieurMoyenne, const double temperatureInterieur←  
Minimum, const double temperatureInterieurMaximum, int heure)
- void [recevoirTraitementHumiditeExterieur](#) (const double humiditeExterieursMoyenne, const double humiditeExterieursMinimum,  
const double humiditeExterieursMaximum, int heure)
- void [recevoirTraitementPressionAtmospherique](#) (const double pressionAtmospheriqueMoyenne, const double pression←  
AtmospheriqueMinimum, const double pressionAtmospheriqueMaximum, int heure)
- void [recevoirTraitementEnsoleillement](#) (const double ensoleillementsMoyen, const double ensoleillementsMinimum, const  
double temperatureExterieurMaximum, int heure)
- void [recevoirAlertesTemperatureInterieur](#) ([SeuilsAlertes](#) seuilsAlertes)
- void [recevoirAlertesTemperatureExterieur](#) ([SeuilsAlertes](#) seuilsAlertes)
- void [recevoirAlertesHumiditeInterieur](#) ([SeuilsAlertes](#) seuilsAlertes)
- void [recevoirAlertesHumiditeExterieur](#) ([SeuilsAlertes](#) seuilsAlertes)
- void [recevoirAlertesPressionAtmospherique](#) ([SeuilsAlertes](#) seuilsAlertes)
- void [recevoirAlertesPoids](#) ([SeuilsAlertes](#))
- void [recevoirAlertesEnsoleillement](#) ([SeuilsAlertes](#), double niveauEnsoleillement)
- void [recevoirAlertesBatterie](#) ([SeuilsAlertes](#), double)

## Signaux

- void [nouvelleMesurePoids](#) (QString poids, QString horodatagePoids)  
*signal vers la classe [InfosPoids](#)*
- void [nouvelleTemperatureInterieur](#) (QString temperatureInterieur, QString horodatage)  
*signal vers la classe [InfosTemperature](#)*
- void [nouvelleHumiditeInterieur](#) (QString humiditeInterieur, QString horodatage)  
*signal vers la classe [InfosHumidite](#)*
- void [nouvelleTemperatureExterieur](#) (QString temperatureExterieur, QString horodatage)  
*signal vers la classe [InfosTemperature](#)*
- void [nouvelleHumiditeExterieur](#) (QString humiditeExterieur, QString horodatage)  
*signal vers la classe [InfosHumidite](#)*
- void [nouvellePressionAtmospherique](#) (QString pressionAtmospherique, QString horodatage)  
*signal vers la classe [InfosPressionAtmospherique](#)*
- void [nouvelleMesureEnsoleillement](#) (QString ensoleillement, QString horodatage)  
*signal vers la classe [InfosEnsoleillement](#)*
- void [nouvelleMesureCharge](#) (QString charge, QString horodatage)
- void [nouvelleMesureCourant](#) (QString courant, QString horodatage)
- void [nouvelleMesureTension](#) (QString tension, QString horodatage)
- void [nouvelleMesureTemperatureInterieurTraite](#) (double temperatureInterieurTraite, QString horodatage)  
*signal vers la classe [Ruchelhm](#)*
- void [nouvelleMesureHumiditeInterieurTraite](#) (double humiditeInterieurTraite, QString horodatage)  
*signal vers la classe [Ruchelhm](#)*
- void [nouvelleMesureTemperatureExterieurTraite](#) (double temperatureExterieur, QString horodatage)  
*signal vers la classe [Ruchelhm](#)*
- void [nouvelleMesureHumiditeExterieurTraite](#) (double humiditeExterieurTraite, QString horodatage)  
*signal vers la classe [Ruchelhm](#)*
- void [nouvelleMesureEnsoleillementTraite](#) (double ensoleillement, QString horodatage)  
*signal vers la classe [Ruchelhm](#)*
- void [nouvellePressionAtmospheriqueTraite](#) (double pressionAtmospherique, QString horodatage)  
*signal vers la classe [Ruchelhm](#)*
- void [nouvelleMesureTension](#) (double tension, QString horodatage)
- void [nouvelleMesureCourant](#) (double courant, QString horodatage)
- void [nouvelleMesureCharge](#) (double charge, QString horodatage)
- void [nouvelleMesurePoids](#) (double poids, QString horodatage)
- void [nouvelleMesureTemperatureInterieurTraiteParHeure](#) ()
- void [nouvelleMesureTemperatureExterieurTraiteParHeure](#) ()
- void [nouvelleMesureHumiditeInterieurTraiteParHeure](#) ()
- void [nouvelleMesureHumiditeExterieurTraiteParHeure](#) ()
- void [nouvelleMesurePressionAtmospheriqueTraiteParHeure](#) ()
- void [nouvelleMesureEnsoleillementTraiteParHeure](#) ()
- void [envoiAlertesTemperatureInterieur](#) ([SeuilsAlertes](#))
- void [envoiAlertesTemperatureExterieur](#) ([SeuilsAlertes](#))
- void [envoiAlertesHumiditeInterieur](#) ([SeuilsAlertes](#))
- void [envoiAlertesHumiditeExterieur](#) ([SeuilsAlertes](#))
- void [envoiAlertesPressionAtmospherique](#) ([SeuilsAlertes](#))
- void [envoiAlertesPoids](#) ([SeuilsAlertes](#))
- void [envoiAlertesEnsoleillement](#) ([SeuilsAlertes](#), double)
- void [envoiAlertesBatterie](#) ([SeuilsAlertes](#), double)
- void [etatClientConnexion](#) (bool connexion)
- void [envoiLocalisationPasserelle](#) (QString longitude, QString latitude)



## Fonctions membres publiques

- [Ruche](#) (QStringList [donneesRucheTTN](#), QObject \*parent=0)
- [~Ruche](#) ()  
*Destructeur de la classe Ruche.*
- [Alertes](#) \* [getAlertes](#) ()
- QString [getIdRuche](#) ()

## Fonctions membres privées

- void [connecterGestionMessageTraite](#) ()
- void [connecterGestionMessageBrut](#) ()  
*methode regroupant les conection des messages brut*
- QString [extraireDonneesBrutes](#) (const QByteArray &message)
- QString [extrairePoids](#) (const QString &donneesBrutes)
- [PortsTTN](#) [recupererNumeroDePort](#) (QByteArray message)  
*recuperer le numero de port*
- void [extraireDonnees](#) ([PortsTTN](#) port, QByteArray message)  
*extait les données*
- QString [recupererDonnees](#) (QByteArray message, QString objetJson, QString sousObjetJson)  
*recuperer les données*
- QString [recupererHorodatage](#) (QByteArray message, QString objetJson, QString sousObjetJson)  
*recuperer l'horodatage*
- void [gererAlertes](#) ()
- void [insérerDonneesPortMesureRuche](#) ()
- void [insérerDonneesPortMesureEnvironnement](#) ()
- void [insérerDonneesPortEnsoleillement](#) ()
- void [insérerDonneesPortPoids](#) ()
- void [insérerDonneesPortBatterie](#) ()
- void [insérerMesureHoraireRuche](#) ()
- void [insérerMesureHoraireEnvironnement](#) ()
- void [insérerMesureHoraireEnsoleillement](#) ()
- void [insérerMesureHorairePoids](#) ()
- void [insérerMesureHoraireBatterie](#) ()
- void [recupererLocalisationPasserelle](#) (QByteArray message)

## Attributs privés

- [InfosEnsoleillement](#) \* [ensoleillementRuche](#)
- [InfosHumidite](#) \* [humiditeRuche](#)
- [InfosPressionAtmospherique](#) \* [pressionAtmospheriqueRuche](#)
- [InfosTemperature](#) \* [temperatureRuche](#)
- [InfosBatterie](#) \* [infosBatterie](#)
- [InfosPoids](#) \* [infosPoids](#)
- [Communication](#) \* [communicationRuche](#)
- [BaseDeDonnees](#) \* [bdd](#)  
*agrégation de l'objet BaseDeDonnees*
- [Alertes](#) \* [alertes](#)
- QStringList [donneesRucheTTN](#)
- [DonneesRuche](#) [donneesRuche](#)
- [MesureHoraireRuche](#) [mesureHoraireRuche](#)
- [MesureHoraireEnvironnement](#) [mesureHoraireEnvironnement](#)
- [MesuresHoraireEnsoleillement](#) [mesuresHorraireEnsoleillement](#)

## 9.36.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 9.36.2 Documentation des constructeurs et destructeur

### 9.36.2.1 Ruche()

```
Ruche::Ruche (
    QStringList donneesRucheTTN,
    QObject * parent = 0 )
```

Références `alertes`, `bdd`, `BDD_NOMBASE`, `BDD_PASSWORD`, `BDD_SERVEUR`, `BDD_USERNAME`, `communicationRuche`, `BaseDeDonnees::connecter()`, `connecterGestionMessageBrut()`, `connecterGestionMessageTraite()`, `donneesRucheTTN`, `ensoleillementRuche`, `BaseDeDonnees::estConnecte()`, `etatClientConnexion()`, `gererAlertes()`, `BaseDeDonnees::getInstance()`, `humiditeRuche`, `infosBatterie`, `infosPoids`, `pressionAtmospheriqueRuche`, `recevoirEtatClientConnexion()`, `recevoirMessage()`, `Alertes::setInfosBatterie()`, `Alertes::setInfosEnsoleillement()`, `Alertes::setInfosHumidite()`, `Alertes::setInfosPoids()`, `Alertes::setInfosPressionAtmospherique()`, `Alertes::setInfosTemperature()`, et `temperatureRuche`.

```
00043                                     : QObject (parent),
    ensoleillementRuche(0), humiditeRuche(0),
    pressionAtmospheriqueRuche(0), temperatureRuche(0),
    infosBatterie(0), infosPoids(0), communicationRuche(0),
    alertes(0), donneesRucheTTN(donneesRucheTTN)
00044 {
00045     qDebug() << Q_FUNC_INFO << "donneesRucheTTN" << donneesRucheTTN;
00046     bdd = BaseDeDonnees::getInstance();
00047     if(!bdd->estConnecte())
00048         bdd->connecter(BDD_NOMBASE, BDD_USERNAME,
    BDD_PASSWORD, BDD_SERVEUR);
00049
00050     infosBatterie = new InfosBatterie(this);
00051     infosPoids = new InfosPoids(this);
00052     ensoleillementRuche = new InfosEnsoleillement(this);
00053     humiditeRuche = new InfosHumidite(this);
00054     pressionAtmospheriqueRuche = new
    InfosPressionAtmospherique(this);
00055     temperatureRuche = new InfosTemperature(this);
00056
00057     communicationRuche = new Communication(donneesRucheTTN);
00058
00059     alertes = new Alertes(donneesRucheTTN.at(0), donneesRucheTTN.at(1), this);
00060
00061     alertes->setInfosBatterie(infosBatterie);
00062     alertes->setInfosPoids(infosPoids);
00063     alertes->setInfosEnsoleillement(
    ensoleillementRuche);
00064     alertes->setInfosHumidite(humiditeRuche);
00065     alertes->setInfosPressionAtmospherique(
    pressionAtmospheriqueRuche);
00066     alertes->setInfosTemperature(temperatureRuche);
00067
00068     connecterGestionMessageBrut();
00069     connecterGestionMessageTraite();
00070     gererAlertes();
00071
00072     connect(communicationRuche, SIGNAL(messageRecu(const QByteArray &, const
    QMqttTopicName &)), this, SLOT(recevoirMessage(const QByteArray &, const QMqttTopicName &)));
00073     connect(communicationRuche, SIGNAL(etatClientConnexion(bool)),
    this, SLOT(recevoirEtatClientConnexion(bool)));
00074 }
```

### 9.36.2.2 ~Ruche()

```
Ruche::~Ruche ( )
```

Références `communicationRuche`, `BaseDeDonnees::destruireInstance()`, `ensoleillementRuche`, `humiditeRuche`, `infosBatterie`, `infosPoids`, `pressionAtmospheriqueRuche`, et `temperatureRuche`.

```
00083 {
00084     delete ensoleillementRuche;
00085     delete humiditeRuche;
00086     delete pressionAtmospheriqueRuche;
00087     delete temperatureRuche;
00088     delete communicationRuche;
00089     delete infosBatterie;
00090     delete infosPoids;
00091     BaseDeDonnees::destruireInstance();
00092     qDebug() << Q_FUNC_INFO;
00093 }
```

## 9.36.3 Documentation des fonctions membres

## 9.36.3.1 connecterGestionMessageBrut()

```
void Ruche::connecterGestionMessageBrut ( ) [private]
```

methode permettant les conection des messages brut vers les classe infos pour traitement

Références [ensoleillementRuche](#), [humiditeRuche](#), [infosBatterie](#), [infosPoids](#), [nouvelleHumiditeExterieur\(\)](#), [nouvelleHumiditeInterieur\(\)](#), [nouvelleMesureCharge\(\)](#), [nouvelleMesureCourant\(\)](#), [nouvelleMesureEnsoleillement\(\)](#), [nouvelleMesurePoids\(\)](#), [nouvelleMesureTension\(\)](#), [nouvellePressionAtmospherique\(\)](#), [nouvelleTemperatureExterieur\(\)](#), [nouvelleTemperatureInterieur\(\)](#), [pressionAtmospheriqueRuche](#), et [temperatureRuche](#).

Référencé par [Ruche\(\)](#).

```
00112 {
00113     qDebug() << Q_FUNC_INFO;
00114     connect(this, SIGNAL(nouvelleTemperatureInterieur(QString, QString)),
00115             temperatureRuche, SLOT(traiterNouvelleTemperatureInterieur(QString, QString)));
00116     connect(this, SIGNAL(nouvelleTemperatureExterieur(QString, QString)),
00117             temperatureRuche, SLOT(traiterNouvelleTemperatureExterieur(QString, QString)));
00118     connect(this, SIGNAL(nouvelleHumiditeInterieur(QString, QString)),
00119             humiditeRuche, SLOT(traiterNouvelleHumiditeInterieur(QString, QString)));
00120     connect(this, SIGNAL(nouvelleHumiditeExterieur(QString, QString)),
00121             humiditeRuche, SLOT(traiterNouvelleHumiditeExterieur(QString, QString)));
00122     connect(this, SIGNAL(nouvelleMesureEnsoleillement(QString, QString)),
00123             ensoleillementRuche, SLOT(traiterNouvelleEnsoleillement(QString, QString)));
00124     connect(this, SIGNAL(nouvellePressionAtmospherique(QString, QString)),
00125             pressionAtmospheriqueRuche, SLOT(traiterNouvellePressionAtmospherique(QString,
00126             QString)));
00127     connect(this, SIGNAL(nouvelleMesureCharge(QString, QString)),
00128             infosBatterie, SLOT(traiterNouvelleMesureCharge(QString, QString)));
00129     connect(this, SIGNAL(nouvelleMesureCourant(QString, QString)),
00130             infosBatterie, SLOT(traiterNouvelleMesureCourant(QString, QString)));
00131     connect(this, SIGNAL(nouvelleMesureTension(QString, QString)),
00132             infosBatterie, SLOT(traiterNouvelleMesureTension(QString, QString)));
00133     connect(this, SIGNAL(nouvelleMesurePoids(QString, QString)),
00134             infosPoids, SLOT(traiterNouveauPoids(QString, QString)));
00135 }
```

## 9.36.3.2 connecterGestionMessageTraite()

```
void Ruche::connecterGestionMessageTraite ( ) [private]
```

Références [ensoleillementRuche](#), [humiditeRuche](#), [infosBatterie](#), [infosPoids](#), [pressionAtmospheriqueRuche](#), [recevoirEnsoleillementTraite\(\)](#), [recevoirMesureChargeTraite\(\)](#), [recevoirMesureCourantTraite\(\)](#), [recevoirMesureHumiditeExterieurTraite\(\)](#), [recevoirMesureHumiditeInterieurTraite\(\)](#), [recevoirMesurePoidsTraite\(\)](#), [recevoirMesureTensionTraite\(\)](#), [recevoirPressionAtmospheriqueTraite\(\)](#), [recevoirTemperatureExterieurTraite\(\)](#), [recevoirTemperatureInterieurTraite\(\)](#), [recevoirTraitementEnsoleillement\(\)](#), [recevoirTraitementHumiditeExterieur\(\)](#), [recevoirTraitementHumiditeInterieur\(\)](#), [recevoirTraitementPressionAtmospherique\(\)](#), [recevoirTraitementTemperatureExterieur\(\)](#), [recevoirTraitementTemperatureInterieur\(\)](#), et [temperatureRuche](#).

Référencé par [Ruche\(\)](#).

```
00133 {
00134     qDebug() << Q_FUNC_INFO;
00135     connect(temperatureRuche, SIGNAL(temperatureInterieurEnvoye(double, QString)), this, SLOT(
00136             recevoirTemperatureInterieurTraite(double, QString)));
00137     connect(temperatureRuche, SIGNAL(temperatureExterieurEnvoye(double, QString)), this, SLOT(
00138             recevoirTemperatureExterieurTraite(double, QString)));
00139     connect(humiditeRuche, SIGNAL(humiditeInterieurEnvoye(double, QString)), this, SLOT(
00140             recevoirMesureHumiditeInterieurTraite(double, QString)));
00141     connect(humiditeRuche, SIGNAL(humiditeExterieurEnvoye(double, QString)), this, SLOT(
00142             recevoirMesureHumiditeExterieurTraite(double, QString)));
00143     connect(pressionAtmospheriqueRuche, SIGNAL(pressionAtmospheriqueEnvoye(double,
00144             QString)), this, SLOT(recevoirPressionAtmospheriqueTraite(double, QString)));
00145 }
```

```

00140 ;
00140     connect(enseleillementRuche, SIGNAL(enseleillementEnvoye(double,QString)),this, SLOT(
00141     recevoirEnseleillementTraite(double,QString));
00141     connect(infosBatterie, SIGNAL(tensionEnvoye(double,QString)),this, SLOT(
00142     recevoirMesureTensionTraite(double,QString));
00142     connect(infosBatterie, SIGNAL(courantEnvoye(double,QString)),this, SLOT(
00143     recevoirMesureCourantTraite(double,QString));
00143     connect(infosBatterie, SIGNAL(chargeEnvoye(double,QString)), this, SLOT(
00144     recevoirMesureChargeTraite(double,QString));
00144     connect(infosPoids, SIGNAL(poidsEnvoye(double,QString)), this, SLOT(
00145     recevoirMesurePoidsTraite(double,QString));
00145
00146     connect(temperatureRuche, SIGNAL(traitementTemperatureInterieurEnvoye(const double,
00146     const double, const double, int)), this, SLOT(recevoirTraitementTemperatureInterieur
00147     (const double,const double,const double, int)));
00147     connect(temperatureRuche, SIGNAL(traitementTemperatureExterieurEnvoye(const double,
00148     const double, const double, int)), this, SLOT(recevoirTraitementTemperatureExterieur
00148     (const double,const double,const double, int)));
00148     connect(humiditeRuche, SIGNAL(traitementHumiditeInterieurEnvoye(double,double,double,int))
00149     , this, SLOT(recevoirTraitementHumiditeInterieur(double,double,double,int)
00149     ));
00149     connect(humiditeRuche, SIGNAL(traitementHumiditeExterieurEnvoye(double,double,double,int))
00150     , this, SLOT(recevoirTraitementHumiditeExterieur(double,double,double,int)
00150     ));
00150     connect(pressionAtmospheriqueRuche, SIGNAL(
00151     traitementPressionAtmospheriqueEnvoye(double,double,double,int)), this, SLOT(
00151     recevoirTraitementPressionAtmospherique(double,double,double,int));
00151     connect(enseleillementRuche, SIGNAL(traitementEnseleillementEnvoye(double,double,
00152     double,int)), this, SLOT(recevoirTraitementEnseleillement(double,double,double,
00152     int)));
00152 }

```

### 9.36.3.3 envoiAlertesBatterie

```

void Ruche::envoiAlertesBatterie (
    SeuilsAlertes ,
    double ) [signal]

```

Référencé par [gererAlertes\(\)](#), et [recevoirAlertesBatterie\(\)](#).

### 9.36.3.4 envoiAlertesEnseleillement

```

void Ruche::envoiAlertesEnseleillement (
    SeuilsAlertes ,
    double ) [signal]

```

Référencé par [gererAlertes\(\)](#), et [recevoirAlertesEnseleillement\(\)](#).

### 9.36.3.5 envoiAlertesHumiditeExterieur

```

void Ruche::envoiAlertesHumiditeExterieur (
    SeuilsAlertes ) [signal]

```

Référencé par [gererAlertes\(\)](#), et [recevoirAlertesHumiditeExterieur\(\)](#).

### 9.36.3.6 envoiAlertesHumiditeInterieur

```

void Ruche::envoiAlertesHumiditeInterieur (
    SeuilsAlertes ) [signal]

```

Référencé par [gererAlertes\(\)](#), et [recevoirAlertesHumiditeInterieur\(\)](#).

#### 9.36.3.7 envoiAlertesPoids

```
void Ruche::envoiAlertesPoids (
    SeuilsAlertes ) [signal]
```

Référencé par [gererAlertes\(\)](#), et [recevoirAlertesPoids\(\)](#).

#### 9.36.3.8 envoiAlertesPressionAtmospherique

```
void Ruche::envoiAlertesPressionAtmospherique (
    SeuilsAlertes ) [signal]
```

Référencé par [gererAlertes\(\)](#), et [recevoirAlertesPressionAtmospherique\(\)](#).

#### 9.36.3.9 envoiAlertesTemperatureExterieur

```
void Ruche::envoiAlertesTemperatureExterieur (
    SeuilsAlertes ) [signal]
```

Référencé par [gererAlertes\(\)](#), et [recevoirAlertesTemperatureExterieur\(\)](#).

#### 9.36.3.10 envoiAlertesTemperatureInterieur

```
void Ruche::envoiAlertesTemperatureInterieur (
    SeuilsAlertes ) [signal]
```

Référencé par [gererAlertes\(\)](#), et [recevoirAlertesTemperatureInterieur\(\)](#).

#### 9.36.3.11 envoiLocalisationPasserelle

```
void Ruche::envoiLocalisationPasserelle (
    QString longitude,
    QString latitude ) [signal]
```

Référencé par [recupererLocalisationPasserelle\(\)](#).

#### 9.36.3.12 etatClientConnexion

```
void Ruche::etatClientConnexion (
    bool connexion ) [signal]
```

Référencé par [recevoirEtatClientConnexion\(\)](#), et [Ruche\(\)](#).

#### 9.36.3.13 extraireDonnees()

```
void Ruche::extraireDonnees (
    PortsTTN port,
    QByteArray message ) [private]
```

extait les données avec le numéro de port spécifié par la méthode [recupérerNomDePort](#)

## Paramètres

<i>port</i>	correspond au numéro de port
<i>message</i>	message reçu émis par le serveur ttn grace au protocole mqtt

Références `DonneesBatterie : :charge`, `DonneesBatterie : :courant`, `DonneesRuche : :donneesBatterie`, `donneesRuche`, `DonneesRuche : :ensoileillement`, `DonneesBatterie : :horodatagecharge`, `DonneesBatterie : :horodatagecourant`, `DonneesRuche : :horodatageEnsoileillement`, `DonneesRuche : :horodatageHumiditeExterieur`, `DonneesRuche : :horodatageHumiditeInterieur`, `DonneesRuche : :horodatagePoids`, `DonneesRuche : :horodatagePressionAtmospherique`, `DonneesRuche : :horodatageTemperatureExterieur`, `DonneesRuche : :horodatageTemperatureInterieur`, `DonneesBatterie : :horodatagetension`, `DonneesRuche : :humiditeExterieur`, `DonneesRuche : :humiditeInterieur`, `insererDonneesPortBatterie()`, `insererDonneesPortEnsoileillement()`, `insererDonneesPortMesureEnvironnement()`, `insererDonneesPortMesureRuche()`, `insererDonneesPortPoids()`, `nouvelleHumiditeExterieur()`, `nouvelleHumiditeInterieur()`, `nouvelleMesureCharge()`, `nouvelleMesureCourant()`, `nouvelleMesureEnsoileillement()`, `nouvelleMesurePoids()`, `nouvelleMesureTension()`, `nouvellePressionAtmospherique()`, `nouvelleTemperatureExterieur()`, `nouvelleTemperatureInterieur()`, `DonneesRuche : :poids`, `portMesureEnergie`, `portMesureEnsoileillement`, `portMesureEnvironnement`, `portMesurePoids`, `portMesureRuche`, `DonneesRuche : :pressionAtmospherique`, `recupererDonnees()`, `recupererHorodatage()`, `recupererLocalisationPasserelle()`, `DonneesRuche : :temperatureExterieur`, `DonneesRuche : :temperatureInterieur`, et `DonneesBatterie : :tension`.

Référencé par `recevoirMessage()`.

```

00246 {
00247     qDebug() << Q_FUNC_INFO << "port=" << port << "message=" << message;
00248     QJsonDocument documentJSON = QJsonDocument::fromJson(message);
00249     QJsonObject objetJSON = documentJSON.object();
00250
00251     // attributs ?
00252     QString donneesBrutes;
00253
00254     switch (port)
00255     {
00256         case portMesureEnergie :
00257             donneesRuche.donneesBatterie.charge =
00258             recupererDonnees(message, "payload_fields", "charge");
00259             donneesRuche.donneesBatterie.courant =
00260             recupererDonnees(message, "payload_fields", "courant");
00261             donneesRuche.donneesBatterie.tension =
00262             recupererDonnees(message, "payload_fields", "tension");
00263             donneesRuche.donneesBatterie.
00264             horodatagecharge = recupererHorodatage(message, "metadata", "time");
00265             donneesRuche.donneesBatterie.
00266             horodatagecourant = recupererHorodatage(message, "metadata", "time");
00267             donneesRuche.donneesBatterie.
00268             horodatagetension = recupererHorodatage(message, "metadata", "time");
00269
00270             emit nouvelleMesureCharge(donneesRuche.
00271             donneesBatterie.charge, donneesRuche.
00272             donneesBatterie.horodatagecharge );
00273             emit nouvelleMesureCourant(donneesRuche.
00274             donneesBatterie.courant, donneesRuche.
00275             donneesBatterie.horodatagecourant);
00276             emit nouvelleMesureTension(donneesRuche.
00277             donneesBatterie.tension, donneesRuche.
00278             donneesBatterie.horodatagetension);
00279             insererDonneesPortBatterie();
00280             recupererLocalisationPasserelle(message);
00281             qDebug() << Q_FUNC_INFO << "charge = " << donneesRuche.
00282             donneesBatterie.charge << "tension = " << donneesRuche.
00283             donneesBatterie.tension << "courant = " << donneesRuche.
00284             donneesBatterie.courant;
00285             break;
00286         case portMesurePoids :
00287             donneesRuche.poids = recupererDonnees(message, "payload_fields"
00288             ,"poids");
00289             donneesRuche.horodatagePoids =
00290             recupererHorodatage(message, "metadata", "time");
00291             //donneesBrutes = extraireDonneesBrutes(message);
00292             //donneesRuche.poids = extrairePoids(donneesBrutes);
00293             qDebug() << Q_FUNC_INFO << "Mesure poids=" << donneesRuche.
00294             poids;
00295             emit nouvelleMesurePoids(donneesRuche.
00296             poids, donneesRuche.horodatagePoids);
00297             insererDonneesPortPoids();
00298             recupererLocalisationPasserelle(message);
00299             break;
00300         case portMesureRuche :
00301             //qDebug() << Q_FUNC_INFO << "payload_fields:{\"temperature\"}=" <<

```

```

    objetJSON.value("payload_fields").toObject().value("temperature");
00283     donneesRuche.temperatureInterieur =
recupererDonnees(message, "payload_fields", "temperature");
00284     qDebug() << Q_FUNC_INFO << "Mesure temperatureInterieur=" <<
donneesRuche.temperatureInterieur;
00285
00286     donneesRuche.horodatageTemperatureInterieur =
recupererHorodatage(message, "metadata", "time");
00287     qDebug() << Q_FUNC_INFO << "Horodatage temperatureInterieur=" <<
donneesRuche.horodatageTemperatureInterieur;
00288     emit nouvelleTemperatureInterieur(
donneesRuche.temperatureInterieur, donneesRuche.
horodatageTemperatureInterieur);
00289     donneesRuche.humiditeInterieur =
recupererDonnees(message, "payload_fields", "humidite");
00290     donneesRuche.horodatageHumiditeInterieur =
recupererHorodatage(message, "metadata", "time");
00291     emit nouvelleHumiditeInterieur(donneesRuche.
humiditeInterieur, donneesRuche.
horodatageHumiditeInterieur);
00292     insererDonneesPortMesureRuche();
00293     recupererLocalisationPasserelle(message);
00294     break;
00295     case portMesureEnvironnement :
00296         donneesRuche.temperatureExterieur =
recupererDonnees(message, "payload_fields", "temperature");
00297         donneesRuche.horodatageTemperatureExterieur =
recupererHorodatage(message, "metadata", "time");
00298         emit nouvelleTemperatureExterieur(
donneesRuche.temperatureExterieur, donneesRuche.
horodatageTemperatureExterieur);
00299
00300         donneesRuche.humiditeExterieur =
recupererDonnees(message, "payload_fields", "humidite");
00301         donneesRuche.horodatageHumiditeExterieur =
recupererHorodatage(message, "metadata", "time");
00302         emit nouvelleHumiditeExterieur(donneesRuche.
humiditeExterieur, donneesRuche.
horodatageHumiditeExterieur);
00303
00304         donneesRuche.pressionAtmospherique =
recupererDonnees(message, "payload_fields", "pression");
00305         donneesRuche.horodatagePressionAtmospherique =
recupererHorodatage(message, "metadata", "time");
00306         emit nouvellePressionAtmospherique(
donneesRuche.pressionAtmospherique,
donneesRuche.horodatagePressionAtmospherique );
00307         insererDonneesPortMesureEnvironnement();
00308         recupererLocalisationPasserelle(message);
00309         break;
00310
00311     case portMesureEnsoleillement :
00312         donneesRuche.enseillement =
recupererDonnees(message, "payload_fields", "enseillement");
00313         donneesRuche.horodatageEnsoleillement =
recupererHorodatage(message, "metadata", "time");
00314         emit nouvelleMesureEnsoleillement(
donneesRuche.enseillement, donneesRuche.
horodatageEnsoleillement);
00315         insererDonneesPortEnsoleillement();
00316         recupererLocalisationPasserelle(message);
00317         break;
00318     default:
00319         break;
00320 }
00321
00322 }
```

### 9.36.3.14 extraireDonneesBrutes()

```

QString Ruche::extraireDonneesBrutes (
    const QByteArray & message ) [private]

00188 {
00189     QJsonDocument documentJSON = QJsonDocument::fromJson(message);
00190     QJsonObject objetJSON = documentJSON.object();
00191
00192     QString messageTraite = objetJSON.value(QString("payload_raw")).toString();
00193     qDebug() << Q_FUNC_INFO << "payload_raw=" << messageTraite;
00194
00195     return messageTraite;
00196 }
```

### 9.36.3.15 extrairePoids()

```
QString Ruche::extrairePoids (
    const QString & donneesBrutes ) [private]

00199 {
00200     QByteArray donneesBrutesPoids;
00201     unsigned int poidsRuche = 0;
00202     unsigned char poidsRucheMSB;
00203     unsigned char poidsRucheLSB;
00204
00205     donneesBrutesPoids = QByteArray::fromBase64(donneesBrutes.toLocal8Bit());
00206     poidsRucheMSB = donneesBrutesPoids.at(0);
00207     poidsRucheLSB = donneesBrutesPoids.at(1);
00208     poidsRuche = ((poidsRucheMSB << 8) | poidsRucheLSB); // poids en grammes
00209
00210     qDebug("s Poids MSB/LSB = %02X%02X", Q_FUNC_INFO, poidsRucheMSB, poidsRucheLSB);
00211     //qDebug() << Q_FUNC_INFO << "poidsRuche=" << poidsRuche;
00212
00213     return QString::number((double)poidsRuche/(double)1000.); // en Kg
00214 }
```

### 9.36.3.16 gererAlertes()

```
void Ruche::gererAlertes ( ) [private]
```

Références [alertes](#), [envoiAlertesBatterie\(\)](#), [envoiAlertesEnsoleillement\(\)](#), [envoiAlertesHumiditeExterieur\(\)](#), [envoiAlertesHumiditeInterieur\(\)](#), [envoiAlertesPoids\(\)](#), [envoiAlertesPressionAtmospherique\(\)](#), [envoiAlertesTemperatureExterieur\(\)](#), [envoiAlertesTemperatureInterieur\(\)](#), [recevoirAlertesBatterie\(\)](#), [recevoirAlertesEnsoleillement\(\)](#), [recevoirAlertesHumiditeExterieur\(\)](#), [recevoirAlertesHumiditeInterieur\(\)](#), [recevoirAlertesPoids\(\)](#), [recevoirAlertesPressionAtmospherique\(\)](#), [recevoirAlertesTemperatureExterieur\(\)](#), et [recevoirAlertesTemperatureInterieur\(\)](#).

Référencé par [Ruche\(\)](#).

```
00155 {
00156     qDebug() << Q_FUNC_INFO;
00157     connect(alertes, SIGNAL(envoiAlertesTemperatureInterieur(
SeuilsAlertes)),this,SLOT(recevoirAlertesTemperatureInterieur(
SeuilsAlertes)));
00158     connect(alertes, SIGNAL(envoiAlertesTemperatureExterieur(
SeuilsAlertes)),this,SLOT(recevoirAlertesTemperatureExterieur(
SeuilsAlertes)));
00159     connect(alertes, SIGNAL(envoiAlertesHumiditeInterieur(
SeuilsAlertes)),this,SLOT(recevoirAlertesHumiditeInterieur(
SeuilsAlertes)));
00160     connect(alertes, SIGNAL(envoiAlertesHumiditeExterieur(
SeuilsAlertes)),this,SLOT(recevoirAlertesHumiditeExterieur(
SeuilsAlertes)));
00161     connect(alertes, SIGNAL(envoiAlertesPressionAtmospherique(
SeuilsAlertes)),this,SLOT(recevoirAlertesPressionAtmospherique(
SeuilsAlertes)));
00162     connect(alertes, SIGNAL(envoiAlertesEnsoleillement(
SeuilsAlertes, double)),this, SLOT(recevoirAlertesEnsoleillement(
SeuilsAlertes, double)));
00163     connect(alertes, SIGNAL(envoiAlertesBatterie(
SeuilsAlertes,double)),this, SLOT(recevoirAlertesBatterie(
SeuilsAlertes, double)));
00164     connect(alertes, SIGNAL(envoiAlertesPoids(
SeuilsAlertes)),this, SLOT(recevoirAlertesPoids(
SeuilsAlertes)));
00165 }
```



## 9.36.3.17 getAlertes()

```
Alertes * Ruche::getAlertes ( )
```

Références [alertes](#).

Référencé par [RucheIhm : :initialiserMarqueurAlerteEnsoleillement\(\)](#), [RucheIhm : :initialiserMarqueurAlerteHumiditeExterieur\(\)](#), [RucheIhm : :initialiserMarqueurAlerteHumiditeInterieur\(\)](#), [RucheIhm : :initialiserMarqueurAlertePression\(\)](#), [RucheIhm : :initialiserMarqueurAlerteTemperatureExterieur\(\)](#), [RucheIhm : :InitialiserMarqueurAlerteTemperatureInterieur\(\)](#), et [RucheIhm : :selectionnerRuche\(\)](#).

```
00096 {
00097     return alertes;
00098 }
```

## 9.36.3.18 getIdRuche()

```
QString Ruche::getIdRuche ( )
```

Références [donneesRucheTTN](#).

Référencé par [RucheIhm : :afficherAlertesBatterie\(\)](#), [RucheIhm : :afficherAlertesEnsoleillement\(\)](#), [RucheIhm : :afficherAlertesHumiditeExterieur\(\)](#), [RucheIhm : :afficherAlertesHumiditeInterieur\(\)](#), [RucheIhm : :afficherAlertesPoids\(\)](#), [RucheIhm : :afficherAlertesPressionAtmospherique\(\)](#), [RucheIhm : :afficherAlertesTemperatureExterieur\(\)](#), et [RucheIhm : :afficherAlertesTemperatureInterieur\(\)](#).

```
00101 {
00102     return donneesRucheTTN.at (0);
00103 }
```

## 9.36.3.19 insererDonneesPortBatterie()

```
void Ruche::insererDonneesPortBatterie ( ) [private]
```

Références [bdd](#), [DonneesBatterie : :charge](#), [DonneesBatterie : :courant](#), [DonneesRuche : :donneesBatterie](#), [donneesRuche](#), [donneesRucheTTN](#), [BaseDeDonnees : :executer\(\)](#), [InfosBatterie : :getHorodatageBatterie\(\)](#), [infosBatterie](#), et [DonneesBatterie : :tension](#).

Référencé par [extraireDonnees\(\)](#).

```
00580 {
00581     QDateTime dateTimePortMesureBatterie = QDateTime::fromString(infosBatterie->
getHorodatageBatterie(), "dd/MM/yyyy HH:mm:ss");
00582     QString requete = "INSERT INTO MesuresEnergie(idRuche, Charge, Tension, Courant, Horodatage) VALUES ('"
+ donneesRucheTTN.at(0) + "','" + donneesRuche.
donneesBatterie.charge + "','" + donneesRuche.
donneesBatterie.tension + "','" + donneesRuche.
donneesBatterie.courant + "','" + dateTimePortMesureBatterie.toString("yyyy-MM-dd
HH:mm:ss") + "')";
00583     bdd->executer(requete);
00584 }
```

### 9.36.3.20 insérerDonneesPortEnsoleillement()

```
void Ruche::insérerDonneesPortEnsoleillement ( ) [private]
```

Références [bdd](#), [donneesRuche](#), [donneesRucheTTN](#), [DonneesRuche : :ensoleillement](#), [ensoleillementRuche](#), [BaseDeDonnees : :executer\(\)](#), et [InfosEnsoleillement : :getHorodatageEnsoleillement\(\)](#).

Référencé par [extraireDonnees\(\)](#).

```
00565 {
00566     QDateTime dateTimePortMesureEnsoleillement = QDateTime::fromString(
        ensoleillementRuche->getHorodatageEnsoleillement(), "
        dd/MM/yyyy HH:mm:ss");
00567     QString requete = "INSERT INTO MesuresEnsoleillement(idRuche, Ensoleillement, Horodatage) VALUES ('" +
        donneesRucheTTN.at(0) + "','" + donneesRuche.
        ensoleillement + "','" + dateTimePortMesureEnsoleillement.toString("yyyy-MM-dd HH:mm:ss") +
        "')";
00568     bdd->executer(requete);
00569
00570 }
```

### 9.36.3.21 insérerDonneesPortMesureEnvironnement()

```
void Ruche::insérerDonneesPortMesureEnvironnement ( ) [private]
```

Références [bdd](#), [donneesRuche](#), [donneesRucheTTN](#), [BaseDeDonnees : :executer\(\)](#), [InfosTemperature : :getHorodatageTemperatureExterieur\(\)](#), [DonneesRuche : :humiditeExterieur](#), [DonneesRuche : :pressionAtmospherique](#), [DonneesRuche : :temperatureExterieur](#), et [temperatureRuche](#).

Référencé par [extraireDonnees\(\)](#).

```
00557 {
00558     QDateTime dateTimePortMesureEnvironnement = QDateTime::fromString(
        temperatureRuche->getHorodatageTemperatureExterieur(), "
        dd/MM/yyyy HH:mm:ss");
00559     qDebug() << Q_FUNC_INFO << "ruche" << donneesRucheTTN.at(0) <<
        dateTimePortMesureEnvironnement.toString("yyyy-MM-dd HH:mm:ss");
00560     QString requete = "INSERT INTO MesuresEnvironnement(idRuche, Temperature, Humidite, Pression,
        Horodatage) VALUES ('" + donneesRucheTTN.at(0) + "','" + donneesRuche.
        temperatureExterieur + "','" + donneesRuche.
        humiditeExterieur + "','" + donneesRuche.
        pressionAtmospherique + "','" + dateTimePortMesureEnvironnement.toString("yyyy-MM-dd
        HH:mm:ss") + "')";
00561     bdd->executer(requete);
00562 }
```

### 9.36.3.22 insérerDonneesPortMesureRuche()

```
void Ruche::insérerDonneesPortMesureRuche ( ) [private]
```

Références [bdd](#), [donneesRuche](#), [donneesRucheTTN](#), [BaseDeDonnees : :executer\(\)](#), [InfosTemperature : :getHorodatageTemperatureInterieur\(\)](#), [InfosHumidite : :getHumiditeInterieur\(\)](#), [InfosTemperature : :getTemperatureInterieur\(\)](#), [DonneesRuche : :humiditeInterieur](#), [humiditeRuche](#), [DonneesRuche : :temperatureInterieur](#), et [temperatureRuche](#).

Référencé par [extraireDonnees\(\)](#).

```
00548 {
00549
00550     QDateTime dateTimePortMesureRuche = QDateTime::fromString(temperatureRuche->
        getHorodatageTemperatureInterieur(), "dd/MM/yyyy HH:mm:ss");
00551     qDebug() << Q_FUNC_INFO << "ruche" << donneesRucheTTN.at(0) << dateTimePortMesureRuche.
        toString("yyyy-MM-dd HH:mm:ss") << "temperatureInterieur" << temperatureRuche->
        getTemperatureInterieur() << "humiditeInterieur" <<
        humiditeRuche->getHumiditeInterieur();
00552     QString requete = "INSERT INTO MesuresRuche (MesuresRuche.idRuche, MesuresRuche.Temperature,
        MesuresRuche.Humidite, MesuresRuche.Horodatage) VALUES ('" + donneesRucheTTN.at(0) + "','" +
        donneesRuche.temperatureInterieur + "','" +
        donneesRuche.humiditeInterieur + "','" + dateTimePortMesureRuche.toString("
        yyyy-MM-dd HH:mm:ss") + "')";
00553     bdd->executer(requete);
00554 }
```

## 9.36.3.23 insererDonneesPortPoids()

```
void Ruche::insererDonneesPortPoids ( ) [private]
```

Références [bdd](#), [donneesRuche](#), [donneesRucheTTN](#), [BaseDeDonnees : :executer\(\)](#), [InfosPoids : :getHorodatagePoids\(\)](#), [infosPoids](#), et [DonneesRuche : :poids](#).

Référencé par [extraireDonnees\(\)](#).

```
00573 {
00574     QDateTime dateTimePortMesurePoids = QDateTime::fromString(infosPoids->
    getHorodatagePoids(), "dd/MM/yyyy HH:mm:ss");
00575     QString requete = "INSERT INTO MesuresPoids(idRuche, Poids, Horodatage) VALUES ('" +
    donneesRucheTTN.at(0) + "','" + donneesRuche.poids + "','" +
    dateTimePortMesurePoids.toString("yyyy-MM-dd HH:mm:ss") + "')";
00576     bdd->executer(requete);
00577 }
```

## 9.36.3.24 insererMesureHoraireBatterie()

```
void Ruche::insererMesureHoraireBatterie ( ) [private]
```

```
00635 {
00636
00637 }
```

## 9.36.3.25 insererMesureHoraireEnsoleillement()

```
void Ruche::insererMesureHoraireEnsoleillement ( ) [private]
```

Références [bdd](#), [donneesRucheTTN](#), [MesuresHoraireEnsoleillement : :ensoleillementsMaximum](#), [MesuresHoraireEnsoleillement : :ensoleillementsMinimum](#), [MesuresHoraireEnsoleillement : :ensoleillementsMoyen](#), [MesuresHoraireEnsoleillement : :ensoleillementsRecues](#), [BaseDeDonnees : :executer\(\)](#), [MesuresHoraireEnsoleillement : :heure](#), et [mesuresHorraireEnsoleillement](#).

Référencé par [recevoirTraitementEnsoleillement\(\)](#).

```
00617 {
00618     if(mesuresHorraireEnsoleillement.
    ensoleillementsRecues)
00619     {
00620         QDate dateMesureEnsoleillement= QDate::currentDate();
00621         QTime heureMesureEnsoleillement(mesuresHorraireEnsoleillement.
    heure, 0);
00622
00623         QString requete = "INSERT INTO MesuresJournalieresEnsoleillement
    (MesuresJournalieresEnsoleillement.idRuche, MesuresJournalieresEnsoleillement.Ensoleillement,
    MesuresJournalieresEnsoleillement.EnsoleillementMin, MesuresJournalieresEnsoleillement.EnsoleillementMax, MesuresJournalieresEn
    MesuresJournalieresEnsoleillement.HeureMesure) VALUES ('" + donneesRucheTTN.at(0) + "','" +
    QString::number(mesuresHorraireEnsoleillement.
    ensoleillementsMoyen) + "','" + QString::number(
    mesuresHorraireEnsoleillement.
    ensoleillementsMinimum) + "','" + QString::number(
    mesuresHorraireEnsoleillement.
    ensoleillementsMaximum) + "','" + dateMesureEnsoleillement.toString("yyyy-MM-dd") + "
    ', '" + heureMesureEnsoleillement.toString("HH:mm:ss") + "')";
00624         bdd->executer(requete);
00625         mesuresHorraireEnsoleillement.
    ensoleillementsRecues = false;
00626     }
00627 }
```

### 9.36.3.26 insererMesureHoraireEnvironnement()

```
void Ruche::insererMesureHoraireEnvironnement ( ) [private]
```

Références `bdd`, `donneesRucheTTN`, `BaseDeDonnees` : `:executer()`, `MesureHoraireEnvironnement` : `:heure`, `MesureHoraireEnvironnement` : `:humiditeExterieurMaximum`, `MesureHoraireEnvironnement` : `:humiditeExterieurMinimum`, `MesureHoraireEnvironnement` : `:humiditeExterieurMoyenne`, `MesureHoraireEnvironnement` : `:humiditesRecues`, `mesureHoraireEnvironnement`, `MesureHoraireEnvironnement` : `:pressionAtmospheriqueMaximum`, `MesureHoraireEnvironnement` : `:pressionAtmospheriqueMinimum`, `MesureHoraireEnvironnement` : `:pressionAtmospheriqueMoyenne`, `MesureHoraireEnvironnement` : `:pressionAtmospheriqueRecues`, `MesureHoraireEnvironnement` : `:temperatureExterieurMaximum`, `MesureHoraireEnvironnement` : `:temperatureExterieurMinimum`, `MesureHoraireEnvironnement` : `:temperatureExterieurMoyenne`, et `MesureHoraireEnvironnement` : `:temperaturesRecues`.

Référencé par `recevoirTraitementHumiditeExterieur()`, `recevoirTraitementPressionAtmospherique()`, et `recevoirTraitementTemperatureExterieur()`.

```
00602 {
00603     if(mesureHoraireEnvironnement.temperaturesRecues &&
        mesureHoraireEnvironnement.humiditesRecues &&
        mesureHoraireEnvironnement.pressionAtmospheriqueRecues)
00604     {
00605         QDate dateMesureEnvironnement = QDate::currentDate();
00606         QTime heureMesureEnvironnement(mesureHoraireEnvironnement.
            heure, 0);
00607
00608         QString requete = "INSERT INTO MesuresJournalieresEnvironnement
            (MesuresJournalieresEnvironnement.idRuche, MesuresJournalieresEnvironnement.Temperature, MesuresJournalieresEnvironnement.TemperatureMax, MesuresJournalieresEnvironnement.Humidite, MesuresJournalieresEnvironnement.HumiditeMin, MesuresJournalieresEnvironnement.HumiditeMax, MesuresJournalieresEnvironnement.Pression, MesuresJournalieresEnvironnement.PressionMin, MesuresJournalieresEnvironnement.PressionMax, MesuresJournalieresEnvironnement.DateMesure, MesuresJournalieresEnvironnement.HeureMesure) VALUES ('" +
            donneesRucheTTN.at(0) + "','" + QString::number(
                mesureHoraireEnvironnement.temperatureExterieurMoyenne)
            + "','" + QString::number(mesureHoraireEnvironnement.
                temperatureExterieurMinimum) + "','" + QString::number(
                mesureHoraireEnvironnement.temperatureExterieurMaximum)
            + "','" + QString::number(mesureHoraireEnvironnement.
                humiditeExterieurMoyenne) + "','" + QString::number(
                mesureHoraireEnvironnement.humiditeExterieurMinimum) + "','"
            + QString::number(mesureHoraireEnvironnement.
                humiditeExterieurMaximum) + "','" + QString::number(
                mesureHoraireEnvironnement.pressionAtmospheriqueMoyenne)
            + "','" + QString::number(mesureHoraireEnvironnement.
                pressionAtmospheriqueMinimum) + "','" + QString::number(
                mesureHoraireEnvironnement.pressionAtmospheriqueMaximum)
            + "','" + dateMesureEnvironnement.toString("yyyy-MM-dd") + "','" + heureMesureEnvironnement.toString("HH:mm:ss") + "')";
00609         bdd->executer(requete);
00610         mesureHoraireEnvironnement.temperaturesRecues = false;
00611         mesureHoraireEnvironnement.humiditesRecues = false;
00612         mesureHoraireEnvironnement.
            pressionAtmospheriqueRecues = false;
00613     }
00614 }
```

### 9.36.3.27 insererMesureHorairePoids()

```
void Ruche::insererMesureHorairePoids ( ) [private]
```

```
00630 {
00631
00632 }
```

## 9.36.3.28 insererMesureHoraireRuche()

```
void Ruche::insererMesureHoraireRuche ( ) [private]
```

Références [bdd](#), [donneesRucheTTN](#), [BaseDeDonnees](#) : [:executer\(\)](#), [MesureHoraireRuche](#) : [:heure](#), [MesureHoraireRuche](#) : [:humiditeInterieurMaximum](#), [MesureHoraireRuche](#) : [:humiditeInterieurMinimum](#), [MesureHoraireRuche](#) : [:humiditeInterieurMoyenne](#), [MesureHoraireRuche](#) : [:humiditesRecues](#), [mesureHoraireRuche](#), [MesureHoraireRuche](#) : [:temperatureInterieurMaximum](#), [MesureHoraireRuche](#) : [:temperatureInterieurMinimum](#), [MesureHoraireRuche](#) : [:temperatureInterieurMoyenne](#), et [MesureHoraireRuche](#) : [:temperaturesRecues](#).

Référencé par [recevoirTraitementHumiditeInterieur\(\)](#), et [recevoirTraitementTemperatureInterieur\(\)](#).

```
00588 {
00589     if(mesureHoraireRuche.temperaturesRecues &&
        mesureHoraireRuche.humiditesRecues)
00590     {
00591         QDate dateMesureRuche = QDate::currentDate();
00592         QTime heureMesureRuche(mesureHoraireRuche.heure, 0);
00593
00594         QString requete = "INSERT INTO MesuresJournalieresRuche (MesuresJournalieresRuche.idRuche,
        MesuresJournalieresRuche.Temperature, MesuresJournalieresRuche.TemperatureMin,
        MesuresJournalieresRuche.TemperatureMax, MesuresJournalieresRuche.Humidite, MesuresJournalieresRuche.HumiditeMin,
        MesuresJournalieresRuche.HumiditeMax, MesuresJournalieresRuche.DateMesure, MesuresJournalieresRuche.HeureMesure) VALUES ('" +
        donneesRucheTTN.at(0) + "',''" + QString::number(mesureHoraireRuche.
        temperatureInterieurMoyenne) + "',''" + QString::number(
        mesureHoraireRuche.temperatureInterieurMinimum) + "',''" +
        QString::number(mesureHoraireRuche.temperatureInterieurMaximum) + "'
        ','" + QString::number(mesureHoraireRuche.
        humiditeInterieurMoyenne) + "',''" + QString::number(
        mesureHoraireRuche.humiditeInterieurMinimum) + "',''" +
        QString::number(mesureHoraireRuche.humiditeInterieurMaximum) + "',''" +
        dateMesureRuche.toString("yyyy-MM-dd") + "',''" + heureMesureRuche.toString("HH:mm:ss") + "')";
00595         bdd->executer(requete);
00596         mesureHoraireRuche.temperaturesRecues = false;
00597         mesureHoraireRuche.humiditesRecues = false;
00598     }
00599 }
```

## 9.36.3.29 nouvelleHumiditeExterieur

```
void Ruche::nouvelleHumiditeExterieur (
    QString humiditeExterieur,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), et [extraireDonnees\(\)](#).

## 9.36.3.30 nouvelleHumiditeInterieur

```
void Ruche::nouvelleHumiditeInterieur (
    QString humiditeInterieur,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), et [extraireDonnees\(\)](#).

## 9.36.3.31 nouvelleMesureCharge [1/2]

```
void Ruche::nouvelleMesureCharge (
    QString charge,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), [extraireDonnees\(\)](#), et [recevoirMesureChargeTraite\(\)](#).

**9.36.3.32 nouvelleMesureCharge** [2/2]

```
void Ruche::nouvelleMesureCharge (
    double charge,
    QString horodatage ) [signal]
```

**9.36.3.33 nouvelleMesureCourant** [1/2]

```
void Ruche::nouvelleMesureCourant (
    QString courant,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), [extraireDonnees\(\)](#), et [recevoirMesureCourantTraite\(\)](#).

**9.36.3.34 nouvelleMesureCourant** [2/2]

```
void Ruche::nouvelleMesureCourant (
    double courant,
    QString horodatage ) [signal]
```

**9.36.3.35 nouvelleMesureEnsoleillement**

```
void Ruche::nouvelleMesureEnsoleillement (
    QString ensoleillement,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), et [extraireDonnees\(\)](#).

**9.36.3.36 nouvelleMesureEnsoleillementTraite**

```
void Ruche::nouvelleMesureEnsoleillementTraite (
    double ensoleillement,
    QString horodatage ) [signal]
```

Référencé par [recevoirEnsoleillementTraite\(\)](#).

**9.36.3.37 nouvelleMesureEnsoleillementTraiteParHeure**

```
void Ruche::nouvelleMesureEnsoleillementTraiteParHeure ( ) [signal]
```

Référencé par [recevoirTraitementEnsoleillement\(\)](#).

**9.36.3.38 nouvelleMesureHumiditeExterieurTraite**

```
void Ruche::nouvelleMesureHumiditeExterieurTraite (
    double humiditeExterieurTraite,
    QString horodatage ) [signal]
```

Référencé par [recevoirMesureHumiditeExterieurTraite\(\)](#).

**9.36.3.39 nouvelleMesureHumiditeExterieurTraiteParHeure**

```
void Ruche::nouvelleMesureHumiditeExterieurTraiteParHeure ( ) [signal]
```

Référencé par [recevoirTraitementHumiditeExterieur\(\)](#).

**9.36.3.40 nouvelleMesureHumiditeInterieurTraite**

```
void Ruche::nouvelleMesureHumiditeInterieurTraite (
    double humiditeInterieurTraite,
    QString horodatage ) [signal]
```

Référencé par [recevoirMesureHumiditeInterieurTraite\(\)](#).

**9.36.3.41 nouvelleMesureHumiditeInterieurTraiteParHeure**

```
void Ruche::nouvelleMesureHumiditeInterieurTraiteParHeure ( ) [signal]
```

Référencé par [recevoirTraitementHumiditeInterieur\(\)](#).

**9.36.3.42 nouvelleMesurePoids** [1/2]

```
void Ruche::nouvelleMesurePoids (
    QString poids,
    QString horodatagePoids ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), [extraireDonnees\(\)](#), et [recevoirMesurePoidsTraite\(\)](#).

**9.36.3.43 nouvelleMesurePoids** [2/2]

```
void Ruche::nouvelleMesurePoids (
    double poids,
    QString horodatage ) [signal]
```

**9.36.3.44 nouvelleMesurePressionAtmospheriqueTraiteParHeure**

```
void Ruche::nouvelleMesurePressionAtmospheriqueTraiteParHeure ( ) [signal]
```

Référencé par [recevoirTraitementPressionAtmospherique\(\)](#).

**9.36.3.45 nouvelleMesureTemperatureExterieurTraite**

```
void Ruche::nouvelleMesureTemperatureExterieurTraite (
    double temperatureExterieur,
    QString horodatage ) [signal]
```

Référencé par [recevoirTemperatureExterieurTraite\(\)](#).

**9.36.3.46 nouvelleMesureTemperatureExterieurTraiteParHeure**

```
void Ruche::nouvelleMesureTemperatureExterieurTraiteParHeure ( ) [signal]
```

Référencé par [recevoirTraitementTemperatureExterieur\(\)](#).

**9.36.3.47 nouvelleMesureTemperatureInterieurTraite**

```
void Ruche::nouvelleMesureTemperatureInterieurTraite (
    double temperatureInterieurTraite,
    QString horodatage ) [signal]
```

Référencé par [recevoirTemperatureInterieurTraite\(\)](#).

**9.36.3.48 nouvelleMesureTemperatureInterieurTraiteParHeure**

```
void Ruche::nouvelleMesureTemperatureInterieurTraiteParHeure ( ) [signal]
```

Référencé par [recevoirTraitementTemperatureInterieur\(\)](#).

**9.36.3.49 nouvelleMesureTension** [1/2]

```
void Ruche::nouvelleMesureTension (
    QString tension,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), [extraireDonnees\(\)](#), et [recevoirMesureTensionTraite\(\)](#).



**9.36.3.50 nouvelleMesureTension** [2/2]

```
void Ruche::nouvelleMesureTension (
    double tension,
    QString horodatage ) [signal]
```

**9.36.3.51 nouvellePressionAtmospherique**

```
void Ruche::nouvellePressionAtmospherique (
    QString pressionAtmospherique,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), et [extraireDonnees\(\)](#).

**9.36.3.52 nouvellePressionAtmospheriqueTraite**

```
void Ruche::nouvellePressionAtmospheriqueTraite (
    double pressionAtmospherique,
    QString horodatage ) [signal]
```

Référencé par [recevoirPressionAtmospheriqueTraite\(\)](#).

**9.36.3.53 nouvelleTemperatureExterieur**

```
void Ruche::nouvelleTemperatureExterieur (
    QString temperatureExterieur,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), et [extraireDonnees\(\)](#).

**9.36.3.54 nouvelleTemperatureInterieur**

```
void Ruche::nouvelleTemperatureInterieur (
    QString temperatureInterieur,
    QString horodatage ) [signal]
```

Référencé par [connecterGestionMessageBrut\(\)](#), et [extraireDonnees\(\)](#).

**9.36.3.55 recevoirAlertesBatterie**

```
void Ruche::recevoirAlertesBatterie (
    SeuilsAlertes seuilsAlertes,
    double charge ) [slot]
```

Références [envoiAlertesBatterie\(\)](#).

Référencé par [gererAlertes\(\)](#).

```
00543 {
00544     emit envoiAlertesBatterie(seuilsAlertes, charge);
00545 }
```

### 9.36.3.56 recevoirAlertesEnsoleillement

```
void Ruche::recevoirAlertesEnsoleillement (
    SeuilsAlertes seuilsAlertes,
    double niveauEnsoleillement ) [slot]
```

Références [envoiAlertesEnsoleillement\(\)](#).

Référencé par [gererAlertes\(\)](#).

```
00538 {
00539     emit envoiAlertesEnsoleillement(seuilsAlertes, niveauEnsoleillement);
00540 }
```

### 9.36.3.57 recevoirAlertesHumiditeExterieur

```
void Ruche::recevoirAlertesHumiditeExterieur (
    SeuilsAlertes seuilsAlertes ) [slot]
```

Références [envoiAlertesHumiditeExterieur\(\)](#).

Référencé par [gererAlertes\(\)](#).

```
00521 {
00522     emit envoiAlertesHumiditeExterieur(seuilsAlertes);
00523 }
```

### 9.36.3.58 recevoirAlertesHumiditeInterieur

```
void Ruche::recevoirAlertesHumiditeInterieur (
    SeuilsAlertes seuilsAlertes ) [slot]
```

Références [envoiAlertesHumiditeInterieur\(\)](#).

Référencé par [gererAlertes\(\)](#).

```
00514 {
00515     qDebug() << Q_FUNC_INFO;
00516
00517     emit envoiAlertesHumiditeInterieur(seuilsAlertes);
00518 }
```

### 9.36.3.59 recevoirAlertesPoids

```
void Ruche::recevoirAlertesPoids (
    SeuilsAlertes seuilsAlertes ) [slot]
```

Références [envoiAlertesPoids\(\)](#).

Référencé par [gererAlertes\(\)](#).

```
00533 {
00534     emit envoiAlertesPoids(seuilsAlertes);
00535 }
```

**9.36.3.60 recevoirAlertesPressionAtmospherique**

```
void Ruche::recevoirAlertesPressionAtmospherique (
    SeuilsAlertes seuilsAlertes ) [slot]
```

Références [envoiAlertesPressionAtmospherique\(\)](#).

Référencé par [gererAlertes\(\)](#).

```
00526 {
00527     qDebug() << Q_FUNC_INFO;
00528
00529     emit envoiAlertesPressionAtmospherique(seuilsAlertes);
00530 }
```

**9.36.3.61 recevoirAlertesTemperatureExterieur**

```
void Ruche::recevoirAlertesTemperatureExterieur (
    SeuilsAlertes seuilsAlertes ) [slot]
```

Références [envoiAlertesTemperatureExterieur\(\)](#).

Référencé par [gererAlertes\(\)](#).

```
00507 {
00508     qDebug() << Q_FUNC_INFO;
00509
00510     emit envoiAlertesTemperatureExterieur(seuilsAlertes);
00511 }
```

**9.36.3.62 recevoirAlertesTemperatureInterieur**

```
void Ruche::recevoirAlertesTemperatureInterieur (
    SeuilsAlertes seuilsAlertes ) [slot]
```

Références [envoiAlertesTemperatureInterieur\(\)](#).

Référencé par [gererAlertes\(\)](#).

```
00500 {
00501     qDebug() << Q_FUNC_INFO << seuilsAlertes;
00502
00503     emit envoiAlertesTemperatureInterieur(seuilsAlertes);
00504 }
```

**9.36.3.63 recevoirEnsoleillementTraite**

```
void Ruche::recevoirEnsoleillementTraite (
    double ensoleillement,
    QString horodatage ) [slot]
```

cette fonction émet la valeur traitée a l'ihm

## Paramètres

<i>ensoleillement</i>	correspond aux mesures d'ensoleillement traitée
<i>horodatage</i>	corespond a horodatage de l'ensoleillement exterieur traitée

Références [alerteEnsoleillement](#), [alertes](#), [Alertes : :appelerLesAlertes\(\)](#), et [nouvelleMesureEnsoleillementTraite\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00492 {
00493     qDebug() << Q_FUNC_INFO;
00494
00495     emit nouvelleMesureEnsoleillementTraite(ensoleillement,horodatage);
00496     alertes->appelerLesAlertes(alerteEnsoleillement);
00497 }
```

## 9.36.3.64 recevoirEtatClientConnexion

```
void Ruche::recevoirEtatClientConnexion (
    bool etat ) [slot]
```

Références [etatClientConnexion\(\)](#).

Référencé par [Ruche\(\)](#).

```
00183 {
00184     emit etatClientConnexion(etat);
00185 }
```

## 9.36.3.65 recevoirMessage

```
void Ruche::recevoirMessage (
    const QByteArray & message,
    const QMqttTopicName & topic ) [slot]
```

slot permettant la reception des données et du nom de topic

## Paramètres

<i>message</i>	message reçu émis par le serveur ttn grace au potocole mqtt
<i>topic</i>	nom du topic emettant la données

Références [extraireDonnees\(\)](#), et [recupererNumeroDePort\(\)](#).

Référencé par [Ruche\(\)](#).

```
00176 {
00177     Q_UNUSED(topic)
00178     PortsTTN port = recupererNumeroDePort(message);
00179     extraireDonnees(port,message);
00180 }
```

**9.36.3.66 recevoirMesureChargeTraite**

```
void Ruche::recevoirMesureChargeTraite (
    double charge,
    QString horodatage ) [slot]
```

Références [alerteBatterie](#), [alertes](#), [Alertes : :appelerLesAlertes\(\)](#), et [nouvelleMesureCharge\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00472 {
00473     emit nouvelleMesureCharge(charge,horodatage);
00474     alertes->appelerLesAlertes(alerteBatterie);
00475 }
```

**9.36.3.67 recevoirMesureCourantTraite**

```
void Ruche::recevoirMesureCourantTraite (
    double courant,
    QString horodatage ) [slot]
```

Références [nouvelleMesureCourant\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00467 {
00468     emit nouvelleMesureCourant(courant,horodatage);
00469 }
```

**9.36.3.68 recevoirMesureHumiditeExterieurTraite**

```
void Ruche::recevoirMesureHumiditeExterieurTraite (
    double humiditeExterieurTraite,
    QString horodatage ) [slot]
```

Références [alerteHumiditeExterieur](#), [alertes](#), [Alertes : :appelerLesAlertes\(\)](#), et [nouvelleMesureHumiditeExterieurTraite\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00443 {
00444     emit nouvelleMesureHumiditeExterieurTraite(humiditeExterieurTraite
, horodatage);
00445     alertes->appelerLesAlertes(alerteHumiditeExterieur);
00446 }
```

### 9.36.3.69 recevoirMesureHumiditeInterieurTraite

```
void Ruche::recevoirMesureHumiditeInterieurTraite (
    double humiditeInterieurTraite,
    QString horodatage ) [slot]
```

Références [alerteHumiditeInterieur](#), [alertes](#), [Alertes : :appelerLesAlertes\(\)](#), et [nouvelleMesureHumiditeInterieurTraite\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00413 {
00414     qDebug() << Q_FUNC_INFO << "humiditeInterieurTraite=" << humiditeInterieurTraite;
00415     alertes->appelerLesAlertes(alerteHumiditeInterieur);
00416     emit nouvelleMesureHumiditeInterieurTraite(humiditeInterieurTraite
, horodatage);
00417 }
```

### 9.36.3.70 recevoirMesurePoidsTraite

```
void Ruche::recevoirMesurePoidsTraite (
    double poids,
    QString horodatage ) [slot]
```

Références [alertePoids](#), [alertes](#), [Alertes : :appelerLesAlertes\(\)](#), et [nouvelleMesurePoids\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00478 {
00479     emit nouvelleMesurePoids(poids,horodatage);
00480     alertes->appelerLesAlertes(alertePoids);
00481 }
```

### 9.36.3.71 recevoirMesureTensionTraite

```
void Ruche::recevoirMesureTensionTraite (
    double tension,
    QString horodatage ) [slot]
```

Références [nouvelleMesureTension\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00462 {
00463     emit nouvelleMesureTension(tension,horodatage);
00464 }
```

### 9.36.3.72 recevoirPressionAtmospheriqueTraite

```
void Ruche::recevoirPressionAtmospheriqueTraite (
    double pressionAtmospherique,
    QString horodatage ) [slot]
```

Références [alertePressionAtmospherique](#), [alertes](#), [Alertes : :appelerLesAlertes\(\)](#), et [nouvellePressionAtmospheriqueTraite\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00456 {
00457     emit nouvellePressionAtmospheriqueTraite(pressionAtmospherique,
00458         horodatage);
00459     alertes->appelerLesAlertes(
00460         alertePressionAtmospherique);
00461 }
```

### 9.36.3.73 recevoirTemperatureExterieurTraite

```
void Ruche::recevoirTemperatureExterieurTraite (
    double temperatureExterieur,
    QString horodatage ) [slot]
```

cette fonction emet la valeur traitée a l'ihm

#### Paramètres

<i>temperatureExterieur</i>	correspond à la temperature extérieure traitée
<i>horodatage</i>	correspond a l'horodatage de la temperature exterieur

Références [alertes](#), [alerteTemperatureExterieur](#), [Alertes : :appelerLesAlertes\(\)](#), et [nouvelleMesureTemperatureExterieurTraite\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00428 {
00429     qDebug() << Q_FUNC_INFO << "temperature Exterieur =" << temperatureExterieur;
00430     emit nouvelleMesureTemperatureExterieurTraite(
00431         temperatureExterieur, horodatage);
00432     alertes->appelerLesAlertes(
00433         alerteTemperatureExterieur);
00434 }
```

### 9.36.3.74 recevoirTemperatureInterieurTraite

```
void Ruche::recevoirTemperatureInterieurTraite (
    double temperatureInterieur,
    QString horodatage ) [slot]
```

cette fonction emet la valeur traitée a l'ihm

#### Paramètres

<i>temperatureInterieur</i>	correspond aux mesures temperature traitée
<i>horodatage</i>	correspond a l'horodatage de la temperature interieur

Références [alertes](#), [alerteTemperatureInterieur](#), [Alertes : :appelerLesAlertes\(\)](#), et [nouvelleMesureTemperatureInterieurTraite\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00397 {
00398     qDebug() << Q_FUNC_INFO << "temperatureInterieur=" << temperatureInterieur;
00399     alertes->appelerLesAlertes(
00400         alerteTemperatureInterieur);
00401     emit nouvelleMesureTemperatureInterieurTraite(
00402         temperatureInterieur, horodatage);
00403 }
```

### 9.36.3.75 recevoirTraitementEnsoleillement

```
void Ruche::recevoirTraitementEnsoleillement (
    const double ensoleillementsMoyen,
    const double ensoleillementsMinimum,
    const double temperatureExterieurMaximum,
    int heure ) [slot]
```

Références [MesuresHoraireEnsoleillement : :ensoleillementsMaximum](#), [MesuresHoraireEnsoleillement : :ensoleillementsMinimum](#), [MesuresHoraireEnsoleillement : :ensoleillementsMoyen](#), [MesuresHoraireEnsoleillement : :ensoleillementsRecues](#), [MesuresHoraireEnsoleillement : :heure](#), [insererMesureHoraireEnsoleillement\(\)](#), [mesuresHorraireEnsoleillement](#), et [nouvelleMesureEnsoleillementTraiteParHeure\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00698 {
00699     mesuresHorraireEnsoleillement.heure = heure;
00700     mesuresHorraireEnsoleillement.
00701     ensoleillementsMoyen = ensoleillementsMoyen;
00702     mesuresHorraireEnsoleillement.
00703     ensoleillementsMinimum = ensoleillementsMinimum;
00704     mesuresHorraireEnsoleillement.
00705     ensoleillementsMaximum = ensoleillementsMaximum;
00706     mesuresHorraireEnsoleillement.
00707     ensoleillementsRecues = true;
00708     insererMesureHoraireEnsoleillement();
00709     emit nouvelleMesureEnsoleillementTraiteParHeure();
00710 }
```

### 9.36.3.76 recevoirTraitementHumiditeExterieur

```
void Ruche::recevoirTraitementHumiditeExterieur (
    const double humiditeExterieursMoyenne,
    const double humiditeExterieursMinimum,
    const double humiditeExterieursMaximum,
    int heure ) [slot]
```

Références [MesureHoraireEnvironnement : :heure](#), [MesureHoraireEnvironnement : :humiditeExterieurMaximum](#), [MesureHoraireEnvironnement : :humiditeExterieurMinimum](#), [MesureHoraireEnvironnement : :humiditeExterieurMoyenne](#), [MesureHoraireEnvironnement : :humiditesRecues](#), [insererMesureHoraireEnvironnement\(\)](#), [mesureHoraireEnvironnement](#), et [nouvelleMesureHumiditeExterieurTraiteParHeure\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00676 {
00677     mesureHoraireEnvironnement.heure = heure;
00678     mesureHoraireEnvironnement.humiditeExterieurMoyenne =
00679     humiditeExterieurMoyenne;
00680     mesureHoraireEnvironnement.humiditeExterieurMinimum =
00681     humiditeExterieurMinimum;
00682     mesureHoraireEnvironnement.humiditeExterieurMaximum =
00683     humiditeExterieurMaximum;
00684     mesureHoraireEnvironnement.humiditesRecues = true;
00685     insererMesureHoraireEnvironnement();
00686     emit nouvelleMesureHumiditeExterieurTraiteParHeure();
00687 }
```



## 9.36.3.77 recevoirTraitementHumiditeInterieur

```
void Ruche::recevoirTraitementHumiditeInterieur (
    const double temperatureInterieurMoyenne,
    const double temperatureInterieurMinimum,
    const double temperatureInterieurMaximum,
    int heure ) [slot]
```

Références [MesureHoraireRuche : :heure](#), [MesureHoraireRuche : :humiditeInterieurMaximum](#), [MesureHoraireRuche : :humiditeInterieurMinimum](#), [MesureHoraireRuche : :humiditeInterieurMoyenne](#), [MesureHoraireRuche : :humiditesRecues](#), [insérerMesureHoraireRuche\(\)](#), [mesureHoraireRuche](#), et [nouvelleMesureHumiditeInterieurTraiteParHeure\(\)](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00664 {
00665     qDebug() << Q_FUNC_INFO << "humiditeInterieurMoyenne" << humiditeInterieurMoyenne << "
humiditeInterieurMinimum" << humiditeInterieurMinimum << "humiditeInterieurMaximum" << humiditeInterieurMaximum << "heure" <<
    heure;
00666     mesureHoraireRuche.heure = heure;
00667     mesureHoraireRuche.humiditeInterieurMoyenne =
humiditeInterieurMoyenne;
00668     mesureHoraireRuche.humiditeInterieurMinimum =
humiditeInterieurMinimum;
00669     mesureHoraireRuche.humiditeInterieurMaximum =
humiditeInterieurMaximum;
00670     mesureHoraireRuche.humiditesRecues = true;
00671     insérerMesureHoraireRuche();
00672     emit nouvelleMesureHumiditeInterieurTraiteParHeure();
00673 }
```

## 9.36.3.78 recevoirTraitementPressionAtmospherique

```
void Ruche::recevoirTraitementPressionAtmospherique (
    const double pressionAtmospheriqueMoyenne,
    const double pressionAtmospheriqueMinimum,
    const double pressionAtmospheriqueMaximum,
    int heure ) [slot]
```

Références [MesureHoraireEnvironnement : :heure](#), [insérerMesureHoraireEnvironnement\(\)](#), [mesureHoraireEnvironnement](#), [nouvelleMesurePressionAtmospheriqueTraiteParHeure\(\)](#), [MesureHoraireEnvironnement : :pressionAtmospheriqueMaximum](#), [MesureHoraireEnvironnement : :pressionAtmospheriqueMinimum](#), [MesureHoraireEnvironnement : :pressionAtmospheriqueMoyenne](#), et [MesureHoraireEnvironnement : :pressionAtmospheriqueRecues](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00687 {
00688     mesureHoraireEnvironnement.heure = heure;
00689     mesureHoraireEnvironnement.
pressionAtmospheriqueMoyenne = pressionAtmospheriqueMoyenne;
00690     mesureHoraireEnvironnement.
pressionAtmospheriqueMinimum = pressionAtmospheriqueMinimum;
00691     mesureHoraireEnvironnement.
pressionAtmospheriqueMaximum = pressionAtmospheriqueMaximum;
00692     mesureHoraireEnvironnement.
pressionAtmospheriqueRecues = true;
00693     insérerMesureHoraireEnvironnement();
00694     emit nouvelleMesurePressionAtmospheriqueTraiteParHeure
    ();
00695 }
```

### 9.36.3.79 recevoirTraitementTemperatureExterieur

```
void Ruche::recevoirTraitementTemperatureExterieur (
    const double temperatureExterieurMoyenne,
    const double temperatureExterieurMinimum,
    const double temperatureExterieurMaximum,
    int heure ) [slot]
```

Références [MesureHoraireEnvironnement : :heure](#), [insererMesureHoraireEnvironnement\(\)](#), [mesureHoraireEnvironnement](#), [nouvelleMesureTemperatureExterieurTraiteParHeure\(\)](#), [MesureHoraireEnvironnement : :temperatureExterieurMaximum](#), [MesureHoraireEnvironnement : :temperatureExterieurMinimum](#), [MesureHoraireEnvironnement : :temperatureExterieurMoyenne](#), et [MesureHoraireEnvironnement : :temperaturesRecues](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00653 {
00654     mesureHoraireEnvironnement.heure = heure;
00655     mesureHoraireEnvironnement.
temperatureExterieurMoyenne = temperatureExterieurMoyenne;
00656     mesureHoraireEnvironnement.
temperatureExterieurMinimum = temperatureExterieurMinimum;
00657     mesureHoraireEnvironnement.
temperatureExterieurMaximum = temperatureExterieurMaximum;
00658     mesureHoraireEnvironnement.temperaturesRecues = true;
00659     insererMesureHoraireEnvironnement();
00660     emit nouvelleMesureTemperatureExterieurTraiteParHeure()
;
00661 }
```

### 9.36.3.80 recevoirTraitementTemperatureInterieur

```
void Ruche::recevoirTraitementTemperatureInterieur (
    const double temperatureInterieurMoyenne,
    const double temperatureInterieurMinimum,
    const double temperatureInterieurMaximum,
    int heure ) [slot]
```

Références [MesureHoraireRuche : :heure](#), [insererMesureHoraireRuche\(\)](#), [mesureHoraireRuche](#), [nouvelleMesureTemperatureInterieurTraiteParHeure\(\)](#), [MesureHoraireRuche : :temperatureInterieurMaximum](#), [MesureHoraireRuche : :temperatureInterieurMinimum](#), [MesureHoraireRuche : :temperatureInterieurMoyenne](#), et [MesureHoraireRuche : :temperaturesRecues](#).

Référencé par [connecterGestionMessageTraite\(\)](#).

```
00641 {
00642     qDebug() << Q_FUNC_INFO << "temperatureInterieurMoyenne" << temperatureInterieurMoyenne << "
temperatureInterieurMinimum" << temperatureInterieurMinimum << "temperatureInterieurMaximum" <<
temperatureInterieurMaximum << "heure" << heure;
00643     mesureHoraireRuche.heure = heure;
00644     mesureHoraireRuche.temperatureInterieurMoyenne =
temperatureInterieurMoyenne;
00645     mesureHoraireRuche.temperatureInterieurMinimum =
temperatureInterieurMinimum;
00646     mesureHoraireRuche.temperatureInterieurMaximum =
temperatureInterieurMaximum;
00647     mesureHoraireRuche.temperaturesRecues = true;
00648     insererMesureHoraireRuche();
00649     emit nouvelleMesureTemperatureInterieurTraiteParHeure()
;
00650 }
```

### 9.36.3.81 recupererDonnees()

```
QString Ruche::recupererDonnees (
    QByteArray message,
    QString objetJson,
    QString sousObjetJson ) [private]
```

Méthode permettant de récupérer les données en extrayant celle ci dans un sous objet Json

## Paramètres

<i>message</i>	message reçu message reçu émis par le serveur ttn grace au potocole mqtt
<i>objetJson</i>	nom de l'objet Json
<i>sousObjetJson</i>	nom du sous objet Json

## Renvoi

Un *QString* correspondant a la donnée

Référencé par [extraireDonnees\(\)](#).

```
00353 {
00354     QString donnees;
00355     QJsonDocument documentJSON = QJsonDocument::fromJson(message);
00356     QJsonObject objetJSON = documentJSON.object();
00357
00358     donnees = QString::number(objetJSON.value(objetJson).toObject().value(sousObjetJson).toDouble());
00359     return donnees;
00360 }
```

## 9.36.3.82 recupererHorodatage()

```
QString Ruche::recupererHorodatage (
    QByteArray message,
    QString objetJson,
    QString sousObjetJson ) [private]
```

Méthode permetant de recuperer l'horodatage en extrayant celle ci dans un sous objet Json

## Paramètres

<i>message</i>	message reçu message reçu émis par le serveur ttn grace au potocole mqtt
<i>objetJson</i>	nom de l'objet Json
<i>sousObjetJson</i>	nom du sous objet Json

## Renvoi

Un *QString* correspondant a l'horodatage

Référencé par [extraireDonnees\(\)](#).

```
00373 {
00374     QString horodatageJSON;
00375     QJsonDocument documentJSON = QJsonDocument::fromJson(message);
00376     QJsonObject objetJSON = documentJSON.object();
00377
00378     horodatageJSON = (objetJSON.value(objetJson).toObject().value(sousObjetJson).toString());
00379
00380     // Formatage FR
00381     // QDateTime dateTimeHorodatage = QDateTime::fromString(horodatageJSON.split(".").at(0),
00382     "yyy-MM-ddTHH:mm:ss");
00383     QDateTime dateTimeHorodatage = QDateTime::fromString(horodatageJSON, Qt::ISODate).toLocalTime();
00384     return dateTimeHorodatage.toString("dd/MM/yyyy HH:mm:ss");
00385 }
```

### 9.36.3.83 recupererLocalisationPasserelle()

```
void Ruche::recupererLocalisationPasserelle (
    QByteArray message ) [private]
```

Références [envoiLocalisationPasserelle\(\)](#).

Référencé par [extraireDonnees\(\)](#).

```
00325 {
00326     QString longitude;
00327     QString latitude;
00328
00329     QJsonDocument documentJSON = QJsonDocument::fromJson(message);
00330     QJsonObject objetJSON = documentJSON.object();
00331
00332     qDebug() << Q_FUNC_INFO << "latitude" << objetJSON.value("metadata").toObject().value("gateways").
    toArray().at(0).toObject().value("latitude").toDouble();
00333     qDebug() << Q_FUNC_INFO << "longitude" << objetJSON.value("metadata").toObject().value("gateways").
    toArray().at(0).toObject().value("longitude").toDouble();
00334
00335     latitude = QString::number(objetJSON.value("metadata").toObject().value("gateways").toArray().at(0).
    toObject().value("latitude").toDouble());
00336     longitude = QString::number(objetJSON.value("metadata").toObject().value("gateways").toArray().at(0).
    toObject().value("longitude").toDouble());
00337
00338     emit envoiLocalisationPasserelle(longitude, latitude);
00339 }
```

### 9.36.3.84 recupererNumeroDePort()

```
PortsTTN Ruche::recupererNumeroDePort (
    QByteArray message ) [private]
```

Méthode permettant de recuperer le nupéro de port en extrayant celle ci dans l'objet Json "ports"

#### Paramètres

<i>message</i>	message reçu émis par le serveur ttn grace au protocole mqtt
----------------	--

#### Renvoie

Un *PortsTTN* correspondant au portsttn extrait

Références [nbPortsTTN](#), et [portInconnu](#).

Référencé par [recevoirMessage\(\)](#).

```
00225 {
00226     QJsonDocument documentJSON = QJsonDocument::fromJson(message);
00227     QJsonObject objetJSON = documentJSON.object();
00228     PortsTTN numeroPort = (PortsTTN) objetJSON.value(QString("port")).toInt();
00229     qDebug() << Q_FUNC_INFO << "numeroPort=" << numeroPort;
00230
00231     // port valide ?
00232     if(numeroPort > portInconnu && numeroPort < nbPortsTTN)
00233         return numeroPort;
00234     return portInconnu;
00235 }
```

### 9.36.4 Documentation des données membres

## 9.36.4.1 alertes

```
Alertes* Ruche::alertes [private]
```

Référencé par [gererAlertes\(\)](#), [getAlertes\(\)](#), [recevoirEnsoleillementTraite\(\)](#), [recevoirMesureChargeTraite\(\)](#), [recevoirMesureHumidite↵ExterieurTraite\(\)](#), [recevoirMesureHumiditeInterieurTraite\(\)](#), [recevoirMesurePoidsTraite\(\)](#), [recevoirPressionAtmospheriqueTraite\(\)](#), [recevoirTemperatureExterieurTraite\(\)](#), [recevoirTemperatureInterieurTraite\(\)](#), et [Ruche\(\)](#).

## 9.36.4.2 bdd

```
BaseDeDonnees* Ruche::bdd [private]
```

Référencé par [insererDonneesPortBatterie\(\)](#), [insererDonneesPortEnsoleillement\(\)](#), [insererDonneesPortMesureEnvironnement\(\)](#), [insererDonneesPortMesureRuche\(\)](#), [insererDonneesPortPoids\(\)](#), [insererMesureHoraireEnsoleillement\(\)](#), [insererMesureHoraire↵Environnement\(\)](#), [insererMesureHoraireRuche\(\)](#), et [Ruche\(\)](#).

## 9.36.4.3 communicationRuche

```
Communication* Ruche::communicationRuche [private]
```

Référencé par [Ruche\(\)](#), et [~Ruche\(\)](#).

## 9.36.4.4 donneesRuche

```
DonneesRuche Ruche::donneesRuche [private]
```

Référencé par [extraireDonnees\(\)](#), [insererDonneesPortBatterie\(\)](#), [insererDonneesPortEnsoleillement\(\)](#), [insererDonneesPortMesure↵Environnement\(\)](#), [insererDonneesPortMesureRuche\(\)](#), et [insererDonneesPortPoids\(\)](#).

## 9.36.4.5 donneesRucheTTN

```
QStringList Ruche::donneesRucheTTN [private]
```

Référencé par [getIdRuche\(\)](#), [insererDonneesPortBatterie\(\)](#), [insererDonneesPortEnsoleillement\(\)](#), [insererDonneesPortMesure↵Environnement\(\)](#), [insererDonneesPortMesureRuche\(\)](#), [insererDonneesPortPoids\(\)](#), [insererMesureHoraireEnsoleillement\(\)](#), [inserer↵MesureHoraireEnvironnement\(\)](#), [insererMesureHoraireRuche\(\)](#), et [Ruche\(\)](#).

## 9.36.4.6 ensoleillementRuche

```
InfosEnsoleillement* Ruche::ensoleillementRuche [private]
```

Référencé par [connecterGestionMessageBrut\(\)](#), [connecterGestionMessageTraite\(\)](#), [insererDonneesPortEnsoleillement\(\)](#), [Ruche\(\)](#), et [~Ruche\(\)](#).

#### 9.36.4.7 humiditeRuche

`InfosHumidite*` Ruche::humiditeRuche [private]

Référencé par `connecterGestionMessageBrut()`, `connecterGestionMessageTraite()`, `insererDonneesPortMesureRuche()`, `Ruche()`, et `~Ruche()`.

#### 9.36.4.8 infosBatterie

`InfosBatterie*` Ruche::infosBatterie [private]

Référencé par `connecterGestionMessageBrut()`, `connecterGestionMessageTraite()`, `insererDonneesPortBatterie()`, `Ruche()`, et `~Ruche()`.

#### 9.36.4.9 infosPoids

`InfosPoids*` Ruche::infosPoids [private]

Référencé par `connecterGestionMessageBrut()`, `connecterGestionMessageTraite()`, `insererDonneesPortPoids()`, `Ruche()`, et `~Ruche()`.

#### 9.36.4.10 mesureHoraireEnvironnement

`MesureHoraireEnvironnement` Ruche::mesureHoraireEnvironnement [private]

Référencé par `insererMesureHoraireEnvironnement()`, `recevoirTraitementHumiditeExterieur()`, `recevoirTraitementPressionAtmospherique()`, et `recevoirTraitementTemperatureExterieur()`.

#### 9.36.4.11 mesureHoraireRuche

`MesureHoraireRuche` Ruche::mesureHoraireRuche [private]

Référencé par `insererMesureHoraireRuche()`, `recevoirTraitementHumiditeInterieur()`, et `recevoirTraitementTemperatureInterieur()`.

#### 9.36.4.12 mesuresHorraireEnsoleillement

`MesuresHorraireEnsoleillement` Ruche::mesuresHorraireEnsoleillement [private]

Référencé par `insererMesureHoraireEnsoleillement()`, et `recevoirTraitementEnsoleillement()`.

## 9.36.4.13 pressionAtmospheriqueRuche

`InfosPressionAtmospherique*` Ruche::pressionAtmospheriqueRuche [private]

Référencé par `connecterGestionMessageBrut()`, `connecterGestionMessageTraite()`, `Ruche()`, et `~Ruche()`.

## 9.36.4.14 temperatureRuche

`InfosTemperature*` Ruche::temperatureRuche [private]

Référencé par `connecterGestionMessageBrut()`, `connecterGestionMessageTraite()`, `insererDonneesPortMesureEnvironnement()`, `insererDonneesPortMesureRuche()`, `Ruche()`, et `~Ruche()`.

La documentation de cette classe a été générée à partir des fichiers suivants :

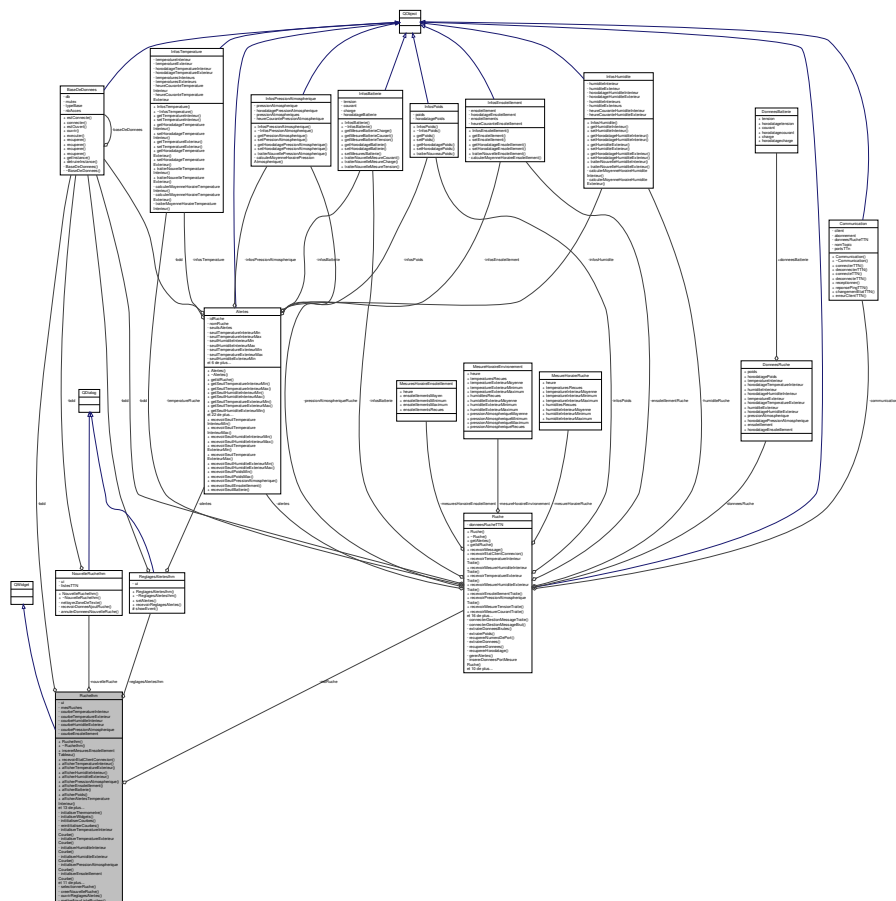
- `ruche.h`
- `ruche.cpp`

## 9.37 Référence de la classe RucheIhm

La classe IHM.

```
#include <rucheIhm.h>
```

Graphe de collaboration de RucheIhm :



## Connecteurs publics

- void [recevoirEtatClientConnexion](#) (bool etat)
- void [afficherTemperatureInterieur](#) (double temperatureInterieur, QString horodatage)  
*affiche temperature interieur à l'aide de widget graphique*
- void [afficherTemperatureExterieur](#) (double temperatureExterieur, QString horodatage)  
*affiche temperature exterieur à l'aide de widget graphique*
- void [afficherHumiditeInterieur](#) (double humiditeInterieur, QString horodatage)  
*affiche humidite interieur à l'aide de widget graphique*
- void [afficherHumiditeExterieur](#) (double humiditeExterieur, QString horodatage)  
*affiche humidite exterieur à l'aide de widget graphique*
- void [afficherPressionAtmospherique](#) (double pressionAtmospherique, QString horodatage)  
*affiche pression atmospherique à l'aide de widget graphique*
- void [afficherEnsoleillement](#) (double ensoleillement, QString horodatage)  
*affiche l'ensoleillement à l'aide de widget graphique*
- void [afficherBatterie](#) (double charge, QString horodatage)
- void [afficherPoids](#) (double poids, QString horodatage)
- void [afficherAlertesTemperatureInterieur](#) ([SeuilsAlertes](#) typeAlerte)
- void [afficherAlertesTemperatureExterieur](#) ([SeuilsAlertes](#) typeAlerte)
- void [afficherAlertesHumiditeInterieur](#) ([SeuilsAlertes](#) typeAlerte)
- void [afficherAlertesHumiditeExterieur](#) ([SeuilsAlertes](#) typeAlerte)
- void [afficherAlertesPressionAtmospherique](#) ([SeuilsAlertes](#) typeAlerte)
- void [afficherAlertesPoids](#) ([SeuilsAlertes](#) typeAlerte)
- void [afficherAlertesEnsoleillement](#) ([SeuilsAlertes](#) typeAlerte, double mesure)
- void [afficherAlertesBatterie](#) ([SeuilsAlertes](#) typeAlerte, double mesure)
- void [afficherMesuresJournalieresRuche](#) ()
- void [afficherMesuresJournalieresEnvironnement](#) ()
- void [afficherMesuresJournalieresEnsoleillement](#) ()
- void [afficherMesuresJournalieresSelectionnee](#) ()
- void [afficherLocalisationPasserelle](#) (QString longitude, QString latitude)
- void [quitter](#) ()

## Signaux

- void [nouvelleDonnesRuche](#) (QString nom, QString prenom, QString email, QString nomTopic)

## Fonctions membres publiques

- [RuchelHm](#) (QWidget \*parent=0)  
*Constructeur de la fenêtre principale.*
- [~RuchelHm](#) ()  
*Destructeur de la fenêtre principale.*
- void [insererMesuresEnsoleillementTableau](#) (QVector< QStringList > mesuresJournalieresEnsoleillement)

## Connecteurs privés

- void [selectionnerRuche](#) (int numeroRuche)
- void [creerNouvelleRuche](#) ()  
*creer une nouvelle ruche*
- void [ouvrirReglagesAlertes](#) ()
- void [mettreAJourListeRuches](#) ()
- void [supprimerRuche](#) ()

## Fonctions membres privées

- void [initialiserThermometre](#) () const  
*initialise les thermometres*
- void [initialiserWidgets](#) ()
- void [initialiserCourbes](#) ()
- void [reinitialiserCourbes](#) ()
- void [initialiserTemperatureInterieurCourbe](#) ()
- void [initialiserTemperatureExterieurCourbe](#) ()
- void [initialiserHumiditeInterieurCourbe](#) ()
- void [initialiserHumiditeExterieurCourbe](#) ()
- void [initialiserPressionAtmospheriqueCourbe](#) ()
- void [initialiserEnsoleillementCourbe](#) ()



- void [InitialiserMarqueurAlerteTemperatureInterieur](#) ()
- void [initialiserMarqueurAlerteHumiditeInterieur](#) ()
- void [initialiserMarqueurAlerteTemperatureExterieur](#) ()
- void [initialiserMarqueurAlerteHumiditeExterieur](#) ()
- void [initialiserMarqueurAlertePression](#) ()
- void [initialiserMarqueurAlerteEnsoleillement](#) ()
- void [renitialiserMarqueurs](#) ()
- void [insererMesuresRucheTableau](#) (QVector< QStringList > mesuresJournalieresRuche)
- void [insererMesuresEnvironnementTableau](#) (QVector< QStringList > mesuresJournalieresEnvironnement)
- void [effacerTableau](#) ()
- void [deconnecterSignaux](#) ()  
*Déconnecte les signaux.*

#### Attributs privés

- Ui : :Ruchelhm \* [ui](#)  
*agrégation de la partie graphique de l'IHM*
- [Ruche](#) \* [maRuche](#)  
*association de l'objet [Ruche](#)*
- [BaseDeDonnees](#) \* [bdd](#)  
*agrégation de l'objet [BaseDeDonnees](#)*
- [NouvelleRuchelhm](#) \* [nouvelleRuche](#)  
*l'ihm pour créer une nouvelle ruche*
- [ReglagesAlertesIhm](#) \* [reglagesAlertesIhm](#)  
*l'ihm pour régler les seuils d'une ruche*
- QVector< QStringList > [mesRuches](#)  
*tableau des informations sur les ruches*
- QwtPlotCurve \* [courbeTemperatureInterieur](#)
- QwtPlotCurve \* [courbeTemperatureExterieur](#)
- QwtPlotCurve \* [courbeHumiditeInterieur](#)
- QwtPlotCurve \* [courbeHumiditeExterieur](#)
- QwtPlotCurve \* [courbePressionAtmospherique](#)
- QwtPlotCurve \* [courbeEnsoleillement](#)

#### 9.37.1 Description détaillée

##### Auteur

Florentin Mellah, Enzo Rossi

##### Version

0.1

#### 9.37.2 Documentation des constructeurs et destructeur

##### 9.37.2.1 Ruchelhm()

```
RucheIhm::RucheIhm (
    QWidget * parent = 0 ) [explicit]
```

##### Paramètres

<i>parent</i>	<a href="#">QObject</a> Adresse de l'objet Qt parent (0 = fenêtre principale)
---------------	---

Références [bdd](#), [BDD\\_NOMBASE](#), [BDD\\_PASSWORD](#), [BDD\\_SERVEUR](#), [BDD\\_USERNAME](#), [BaseDeDonnees](#) : [:connecter\(\)](#), [creerNouvelleRuche\(\)](#), [BaseDeDonnees](#) : [:estConnecte\(\)](#), [BaseDeDonnees](#) : [:getInstance\(\)](#), [initialiserWidgets\(\)](#), [initialiserCourbes\(\)](#),

`mettreAJourListeRuches()`, `nouvelleRuche`, `ouvrirReglagesAlertes()`, `quitter()`, `reglagesAlertesIhm`, `selectionnerRuche()`, `supprimerRuche()`, et `ui`.

```
00040                                     : QWidget(parent), ui(new Ui::RucheIhm),
maRuche(0), bdd(0), nouvelleRuche(0), reglagesAlertesIhm(0)
00041 {
00042     qDebug() << Q_FUNC_INFO;
00043     // Initialisation de l'IHM
00044     ui->setupUi(this);
00045
00046     // Initialisation des widgets
00047     initialiserCourbes();
00048     initialiserWidgets();
00049
00050     // Affichage plein écran
00051     const int width = qApp->desktop()->availableGeometry(this).width(); // ou : qApp->desktop()->width()
00052     const int height = qApp->desktop()->availableGeometry(this).height(); // ou : qApp->desktop()->height()
00053     resize(width, height);
00054
00055     // Ajout de l'action Quitter
00056     QAction *actionQuitter = new QAction("&Quitter", this);
00057     actionQuitter->setShortcut(QKeySequence(QKeySequence::Quit)); // Ctrl+Q
00058     addAction(actionQuitter);
00059
00060     // Connexion signaux/slots
00061     connect(actionQuitter, SIGNAL(triggered()), this, SLOT(quitter()));
00062     connect(ui->comboBoxChoixRuche, SIGNAL(currentIndexChanged(int)), this, SLOT(
selectionnerRuche(int)));
00063     connect(ui->pushButtonNouvelleRuche, SIGNAL(clicked()), this, SLOT(
creerNouvelleRuche()));
00064     connect(ui->pushButtonReglages, SIGNAL(clicked()), this, SLOT(
ouvrirReglagesAlertes()));
00065     connect(ui->pushButtonSuppressionRuche, SIGNAL(clicked()), this, SLOT(
supprimerRuche()));
00066
00067     bdd = BaseDeDonnees::getInstance();
00068     if(!bdd->estConnecte())
00069         bdd->connecter(BDD_NOMBASE, BDD_USERNAME,
BDD_PASSWORD, BDD_SERVEUR);
00070
00071     nouvelleRuche = new NouvelleRucheIhm(this);
00072     reglagesAlertesIhm = new ReglagesAlertesIhm(this);
00073
00074     mettreAJourListeRuches();
00075     //ui->comboBoxChoixRuche->setCurrentIndex(1);
00076 }
```

### 9.37.2.2 ~RucheIhm()

`RucheIhm::~RucheIhm ( )`

Références `BaseDeDonnees::destruireInstance()`, et `ui`.

```
00085 {
00086     delete ui;
00087     BaseDeDonnees::destruireInstance();
00088     qDebug() << Q_FUNC_INFO;
00089 }
```

### 9.37.3 Documentation des fonctions membres

## 9.37.3.1 afficherAlertesBatterie

```
void RucheIhm::afficherAlertesBatterie (
    SeuilsAlertes typeAlerte,
    double mesure ) [slot]
```

Références [alerteBatterie](#), [bdd](#), [bon](#), [BaseDeDonnees](#) : [:executer\(\)](#), [Ruche](#) : [:getIdRuche\(\)](#), [maRuche](#), [tropBas](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00635 {
00636     qDebug() << Q_FUNC_INFO << "SeuilsAlertes " << typeAlerte << "mesure " << mesure;
00637
00638     QString requete;
00639     QDateTime maintenant = QDateTime::currentDateTime();
00640
00641     if(typeAlerte == tropBas && mesure > 0 && mesure < 25)
00642     {
00643         ui->labelAlerteBatterie->setText("<strong>Batterie faible</strong>");
00644         QPixmap imageBatterie(":/images/images/batteryChargeMin.png");
00645         ui->labelAlerteImageBatterie->setPixmap(imageBatterie);
00646         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alerteBatterie) + "', 'Batterie Faible', '" + maintenant.toString("yyyy-MM-dd HH:mm:ss") + "
')";
00647         bdd->executer(requete);
00648     }
00649     else if(typeAlerte == bon)
00650     {
00651         if(mesure >= 75 && mesure <= 100)
00652         {
00653             ui->labelAlerteBatterie->setText("<strong>Batterie</strong>");
00654             QPixmap imageBatterie(":/images/images/batteryChargeMax.png");
00655             ui->labelAlerteImageBatterie->setPixmap(imageBatterie);
00656         }
00657         if(mesure >= 50 && mesure < 75)
00658         {
00659             ui->labelAlerteBatterie->setText("<strong>Batterie</strong>");
00660             QPixmap imageBatterie(":/images/images/batteryChargeBonne.png");
00661             ui->labelAlerteImageBatterie->setPixmap(imageBatterie);
00662         }
00663         if(mesure >= 25 && mesure < 50)
00664         {
00665             ui->labelAlerteBatterie->setText("<strong>Batterie</strong>");
00666             QPixmap imageBatterie(":/images/images/batteryChargeMoitie.png");
00667             ui->labelAlerteImageBatterie->setPixmap(imageBatterie);
00668         }
00669     }
00670 }
```

## 9.37.3.2 afficherAlertesEnsoleillement

```
void RucheIhm::afficherAlertesEnsoleillement (
    SeuilsAlertes typeAlerte,
    double mesure ) [slot]
```

Références [alerteEnsoleillement](#), [bdd](#), [bon](#), [BaseDeDonnees](#) : [:executer\(\)](#), [Ruche](#) : [:getIdRuche\(\)](#), [maRuche](#), [tropHaut](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00589 {
00590     qDebug() << Q_FUNC_INFO << "SeuilsAlertes " << typeAlerte << "mesure " << mesure;
00591
00592     QString requete;
00593     QDateTime maintenant = QDateTime::currentDateTime();
00594
00595     if(typeAlerte == tropHaut)
00596     {
00597         ui->labelAlerteEnsoleillement->setText("<strong>Ensoleillement Elevé</strong>");
00598         QPixmap imageEnsoleillement(":/images/images/alerteEnsoleillement.png");
00599         ui->labelAlerteImageEnsoleillement->setPixmap(imageEnsoleillement);
00600         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
```

```

    alerteEnsoleillement) + "', 'Ensoleillement Elevé'," + maintenant.toString("yyyy-MM-dd
    HH:mm:ss") + "'");
00601     bdd->executer(requete);
00602 }
00603 else if(typeAlerte == bon)
00604 {
00605     if(mesure >=800 && mesure <= 1800)
00606     {
00607         ui->labelAlerteEnsoleillement->setText("<strong>Ensoleillé</strong>");
00608         QPixmap imageEnsoleillement(":/images/images/soleil.png");
00609         ui->labelAlerteImageEnsoleillement->setPixmap(imageEnsoleillement);
00610     }
00611     if(mesure >= 400 && mesure < 800)
00612     {
00613         ui->labelAlerteEnsoleillement->setText("<strong>Ensoleillé dans l'ensemble</strong>");
00614         QPixmap imageEnsoleillement(":/images/images/soleilCouvert.png");
00615         ui->labelAlerteImageEnsoleillement->setPixmap(imageEnsoleillement);
00616     }
00617     if(mesure > 40 && mesure < 400)
00618     {
00619         ui->labelAlerteEnsoleillement->setText("<strong>Couvert</strong>");
00620         QPixmap imageEnsoleillement(":/images/images/nuageux.png");
00621         ui->labelAlerteImageEnsoleillement->setPixmap(imageEnsoleillement);
00622         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
    Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
    alerteEnsoleillement) + "', 'Couvert'," + maintenant.toString("yyyy-MM-dd HH:mm:ss")
    + "'");
00623         bdd->executer(requete);
00624     }
00625     if(mesure <= 40)
00626     {
00627         ui->labelAlerteEnsoleillement->setText("<strong>Nuit</strong>");
00628         QPixmap imageEnsoleillement(":/images/images/moon.png");
00629         ui->labelAlerteImageEnsoleillement->setPixmap(imageEnsoleillement);
00630     }
00631 }
00632 }

```

### 9.37.3.3 afficherAlertesHumiditeExterieur

```

void RucheIhm::afficherAlertesHumiditeExterieur (
    SeuilsAlertes typeAlerte ) [slot]

```

Références [alerteHumiditeExterieur](#), [bdd](#), [bon](#), [BaseDeDonnees](#) : [:executer\(\)](#), [Ruche](#) : [:getIdRuche\(\)](#), [maRuche](#), [tropBas](#), [tropHaut](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```

00498 {
00499     {
00500         qDebug() << Q_FUNC_INFO << "SeuilsAlertes " << typeAlerte;
00501         QString requete;
00502         QDateTime maintenant = QDateTime::currentDateTime();
00503
00504         if(typeAlerte == tropHaut)
00505         {
00506             ui->labelAlerteHumiditeExterieur->setText("<strong>Humidité Ext. Elevée</strong>");
00507             QPixmap imageHumiditeExterieur(":/images/images/humidityHaute.png");
00508             ui->labelAlerteImageHumiditeExterieur->setPixmap(imageHumiditeExterieur);
00509             requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
    Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
    alerteHumiditeExterieur) + "', 'Humidité Extérieur Elevée'," + maintenant.toString(
    "yyyy-MM-dd HH:mm:ss") + "'");
00510             bdd->executer(requete);
00511         }
00512         else if(typeAlerte == tropBas)
00513         {
00514             ui->labelAlerteHumiditeExterieur->setText("<strong>Humidité Ext. Basse</strong>");
00515             QPixmap imageHumiditeExterieur(":/images/images/humidityHaute.png");
00516             ui->labelAlerteImageHumiditeExterieur->setPixmap(imageHumiditeExterieur);
00517             requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
    Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
    alerteHumiditeExterieur) + "', 'Humidité Extérieur Basse'," + maintenant.toString("
    yyyy-MM-dd HH:mm:ss") + "'");
00518             bdd->executer(requete);
00519         }
00520         else if(typeAlerte == bon)
00521         {
00522             ui->labelAlerteHumiditeExterieur->setText("<strong>Humidité Ext. Normale</strong>");
00523             QPixmap imageHumiditeExterieur(":/images/images/humiditeNormal.png");
00524             ui->labelAlerteImageHumiditeExterieur->setPixmap(imageHumiditeExterieur);
00525         }
00526     }
00527 }

```

## 9.37.3.4 afficherAlertesHumiditeInterieur

```
void RucheIhm::afficherAlertesHumiditeInterieur (
    SeuilsAlertes typeAlerte ) [slot]
```

Références [alerteHumiditeInterieur](#), [bdd](#), [bon](#), [BaseDeDonnees : :executer\(\)](#), [Ruche : :getIdRuche\(\)](#), [maRuche](#), [tropBas](#), [tropHaut](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00468 {
00469     qDebug() << Q_FUNC_INFO << "SeuilsAlertes " << typeAlerte;
00470     QString requete;
00471     QDateTime maintenant = QDateTime::currentDateTime();
00472
00473     if(typeAlerte == tropHaut)
00474     {
00475         ui->labelAlerteHumiditeInterieur->setText("<strong>Humidité Int. Elevée</strong>");
00476         QPixmap imageHumiditeInterieur(":/images/images/humidityHaute.png");
00477         ui->labelAlerteImageHumiditeInterieur->setPixmap(imageHumiditeInterieur);
00478         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alerteHumiditeInterieur) + "', 'Humidité Intérieur Elevée','" + maintenant.toString(
"yyyy-MM-dd HH:mm:ss") + "')";
00479         bdd->executer(requete);
00480     }
00481     else if(typeAlerte == tropBas)
00482     {
00483         ui->labelAlerteHumiditeInterieur->setText("<strong>Humidité Int. Basse</strong>");
00484         QPixmap imageHumiditeInterieur(":/images/images/humidityHaute.png");
00485         ui->labelAlerteImageHumiditeInterieur->setPixmap(imageHumiditeInterieur);
00486         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alerteHumiditeInterieur) + "', 'Humidité Intérieur Basse','" + maintenant.toString(
"yyyy-MM-dd HH:mm:ss") + "')";
00487         bdd->executer(requete);
00488     }
00489     else if(typeAlerte == bon)
00490     {
00491         ui->labelAlerteHumiditeInterieur->setText("<strong>Humidité Int. Normale</strong>");
00492         QPixmap imageHumiditeInterieur(":/images/images/humiditeNormal.png");
00493         ui->labelAlerteImageHumiditeInterieur->setPixmap(imageHumiditeInterieur);
00494     }
00495 }
```

## 9.37.3.5 afficherAlertesPoids

```
void RucheIhm::afficherAlertesPoids (
    SeuilsAlertes typeAlerte ) [slot]
```

Références [alertePoids](#), [bdd](#), [bon](#), [BaseDeDonnees : :executer\(\)](#), [Ruche : :getIdRuche\(\)](#), [maRuche](#), [tropBas](#), [tropHaut](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00560 {
00561     qDebug() << Q_FUNC_INFO << "SeuilsAlertes " << typeAlerte;
00562     QString requete;
00563     QDateTime maintenant = QDateTime::currentDateTime();
00564
00565     if(typeAlerte == tropHaut)
00566     {
00567         ui->labelAlertePoids->setText("<strong>Poids Elevé</strong>");
00568         QPixmap imagePoids(":/images/images/poidsAlerte.png");
00569         ui->labelAlerteImagePoids->setPixmap(imagePoids);
00570         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alertePoids) + "', 'Poids Elevé','" + maintenant.toString("yyyy-MM-dd HH:mm:ss") + "')";
00571         bdd->executer(requete);
00572     }
00573     else if(typeAlerte == tropBas)
```

```

00574     {
00575         ui->labelAlertePoids->setText("<strong>Poids Bas</strong>");
00576         QPixmap imagePoids(":/images/images/poidsAlerte.png");
00577         ui->labelAlerteImagePoids->setPixmap(imagePoids);
00578         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alertePoids) + "', 'Poids Bas', '" + maintenant.toString("yyyy-MM-dd HH:mm:ss") + "')";
00579         bdd->executer(requete);
00580     }
00581     else if(typeAlerte == bon)
00582     {
00583         ui->labelAlertePoids->setText("<strong>Poids Normal</strong>");
00584         QPixmap imagePoids(":/images/images/poidsnormal.png");
00585         ui->labelAlerteImagePoids->setPixmap(imagePoids);
00586     }
00587 }

```

### 9.37.3.6 afficherAlertesPressionAtmospherique

```

void RucheIhm::afficherAlertesPressionAtmospherique (
    SeuilsAlertes typeAlerte ) [slot]

```

Références [alertePressionAtmospherique](#), [bdd](#), [bon](#), [BaseDeDonnees](#) : [:executer\(\)](#), [Ruche](#) : [:getIdRuche\(\)](#), [maRuche](#), [tropBas](#), [tropHaut](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```

00530 {
00531     qDebug() << Q_FUNC_INFO << "SeuilsAlertes " << typeAlerte;
00532     QString requete;
00533     QDateTime maintenant = QDateTime::currentDateTime();
00534     if(typeAlerte == tropHaut)
00535     {
00536         ui->labelAlertePressionAtmospherique->setText("<strong>Pression Atmospherique Elevée</strong>");
00537         QPixmap imagePressionAtmospherique(":/images/images/pressionAlerte.png");
00538         ui->labelAlerteImagePressionAtmospherique->setPixmap(imagePressionAtmospherique);
00539         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alertePressionAtmospherique) + "', 'Pression Atmosphérique Elevée', '" +
maintenant.toString("yyyy-MM-dd HH:mm:ss") + "')";
00541         bdd->executer(requete);
00542     }
00543     else if(typeAlerte == tropBas)
00544     {
00545         ui->labelAlertePressionAtmospherique->setText("<strong>Pression Atmospherique Basse</strong>");
00546         QPixmap imagePressionAtmospherique(":/images/images/pressionAlerte.png");
00547         ui->labelAlerteImagePressionAtmospherique->setPixmap(imagePressionAtmospherique);
00548         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alertePressionAtmospherique) + "', 'Pression Atmosphérique Basse', '" +
maintenant.toString("yyyy-MM-dd HH:mm:ss") + "')";
00549         bdd->executer(requete);
00550     }
00551     else if(typeAlerte == bon)
00552     {
00553         ui->labelAlertePressionAtmospherique->setText("<strong>Pression Atmospherique Normale</strong>");
00554         QPixmap imagePressionAtmospherique(":/images/images/pressionNormal.png");
00555         ui->labelAlerteImagePressionAtmospherique->setPixmap(imagePressionAtmospherique);
00556     }
00557 }

```

### 9.37.3.7 afficherAlertesTemperatureExterieur

```

void RucheIhm::afficherAlertesTemperatureExterieur (
    SeuilsAlertes typeAlerte ) [slot]

```

Références [alerteTemperatureExterieur](#), [bdd](#), [bon](#), [BaseDeDonnees](#) : [:executer\(\)](#), [Ruche](#) : [:getIdRuche\(\)](#), [maRuche](#), [tropBas](#), [tropHaut](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```

00430 {
00431     qDebug() << Q_FUNC_INFO << "SeuilsAlertes " << typeAlerte;
00432
00433     QPalette paletteCouleur = ui->thermoTemperatureExterieur->palette();
00434     QString requete;
00435     QDateTime maintenant = QDateTime::currentDateTime();
00436
00437     if(typeAlerte == tropHaut)
00438     {
00439         paletteCouleur.setColor(QPalette::ButtonText, QColor(255,0,0));
00440         ui->thermoTemperatureExterieur->setPalette(paletteCouleur);
00441         ui->labelAlerteTemperatureExterieur->setText("<strong>Température Ext. Elevée</strong>");
00442         QPixmap imageTemperatureExterieur(":/images/images/temperatureHaute.png");
00443         ui->labelImageTemperatureExterieur->setPixmap(imageTemperatureExterieur);
00444         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alerteTemperatureExterieur) + "', 'Température Extérieur Elevée', '" + maintenant.
toString("yyyy-MM-dd HH:mm:ss") + "')";
00445         bdd->executer(requete);
00446     }
00447     else if(typeAlerte == tropBas)
00448     {
00449         paletteCouleur.setColor(QPalette::ButtonText, QColor(135,206,250));
00450         ui->thermoTemperatureExterieur->setPalette(paletteCouleur);
00451         ui->labelAlerteTemperatureExterieur->setText("<strong>Température Ext. Basse</strong>");
00452         QPixmap imageTemperatureExterieur(":/images/images/temperatureBasseAlerte.png");
00453         ui->labelImageTemperatureExterieur->setPixmap(imageTemperatureExterieur);
00454         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alerteTemperatureExterieur) + "', 'Température Extérieur Basse', '" + maintenant.
toString("yyyy-MM-dd HH:mm:ss") + "')";
00455         bdd->executer(requete);
00456     }
00457     else if(typeAlerte == bon)
00458     {
00459         paletteCouleur.setColor(QPalette::ButtonText, QColor(0,128,0));
00460         ui->thermoTemperatureExterieur->setPalette(paletteCouleur);
00461         ui->labelAlerteTemperatureExterieur->setText("<strong>Température Ext. Normale</strong>");
00462         QPixmap imageTemperatureExterieur(":/images/images/temperatureNormal.png");
00463         ui->labelImageTemperatureExterieur->setPixmap(imageTemperatureExterieur);
00464     }
00465 }

```

### 9.37.3.8 afficherAlertesTemperatureInterieur

```

void RucheIhm::afficherAlertesTemperatureInterieur (
    SeuilsAlertes typeAlerte ) [slot]

```

Références [alerteTemperatureInterieur](#), [bdd](#), [bon](#), [BaseDeDonnees : :executer\(\)](#), [Ruche : :getIdRuche\(\)](#), [maRuche](#), [tropBas](#), [tropHaut](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```

00393 {
00394     qDebug() << Q_FUNC_INFO << "SeuilsAlertes " << typeAlerte;
00395     QPalette paletteCouleur = ui->thermoTemperatureInterieur->palette();
00396     QString requete;
00397     QDateTime maintenant = QDateTime::currentDateTime();
00398
00399     if(typeAlerte == tropHaut)
00400     {
00401         paletteCouleur.setColor(QPalette::ButtonText, QColor(255,0,0));
00402         ui->thermoTemperatureInterieur->setPalette(paletteCouleur);
00403         ui->labelAlerteTemperatureInterieur->setText("<strong>Température Int. Elevée</strong>");
00404         QPixmap imageTemperatureInterieur(":/images/images/temperatureHaute.png");
00405         ui->labelAlerteImageTemperature->setPixmap(imageTemperatureInterieur);
00406         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alerteTemperatureInterieur) + "', 'Température Intérieur Elevée', '" + maintenant.
toString("yyyy-MM-dd HH:mm:ss") + "')";
00407         bdd->executer(requete);
00408     }
00409     else if(typeAlerte == tropBas)
00410     {
00411         paletteCouleur.setColor(QPalette::ButtonText, QColor(135,206,250));
00412         ui->thermoTemperatureInterieur->setPalette(paletteCouleur);
00413         ui->labelAlerteTemperatureInterieur->setText("<strong>Température Int. Basse</strong>");
00414         QPixmap imageTemperatureInterieur(":/images/images/temperatureBasseAlerte.png");
00415         ui->labelAlerteImageTemperature->setPixmap(imageTemperatureInterieur);

```

```

00416         requete = "INSERT INTO Alertes (Alertes.idRuche, Alertes.idType, Alertes.Description,
Alertes.Horodatage) VALUES ('" + maRuche->getIdRuche() + "', '" + QString::number(
alerteTemperatureInterieur) + "', 'Température Intérieur Basse', '" + maintenant.
toString("yyyy-MM-dd HH:mm:ss") + "')";
00417         bdd->executer(requete);
00418     }
00419     else if(typeAlerte == bon)
00420     {
00421         paletteCouleur.setColor(QPalette::ButtonText, QColor(0,128,0));
00422         ui->thermoTemperatureInterieur->setPalette(paletteCouleur);
00423         ui->labelAlerteTemperatureInterieur->setText("<strong>Température Int. Normale</strong>");
00424         QPixmap imageTemperatureInterieur(":/images/images/temperatureNormal.png");
00425         ui->labelAlerteImageTemperature->setPixmap(imageTemperatureInterieur);
00426     }
00427 }

```

### 9.37.3.9 afficherBatterie

```

void RucheIhm::afficherBatterie (
    double charge,
    QString horodatage ) [slot]

```

Références [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```

00316 {
00317     qDebug() << Q_FUNC_INFO << "niveau de batterie = " << charge;
00318     ui->lcdNumberBatterie->display(charge);
00319     ui->groupBoxTemperatures->setTitle("Mesures : " + horodatage);
00320 }

```

### 9.37.3.10 afficherEnsoleillement

```

void RucheIhm::afficherEnsoleillement (
    double ensoleillement,
    QString horodatage ) [slot]

```

#### Paramètres

<i>ensoleillement</i>	l'ensoleillement extérieur de la ruche
<i>horodatage</i>	QString

Références [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```

00386 {
00387     qDebug() << Q_FUNC_INFO << "ensoleillement:" << ensoleillement;
00388     ui->lcdNumberEnsoleillement->display(ensoleillement);
00389     ui->groupBoxTemperatures->setTitle("Mesures : " + horodatage);
00390 }

```

### 9.37.3.11 afficherHumiditeExterieur

```

void RucheIhm::afficherHumiditeExterieur (
    double humiditeExterieur,
    QString horodatage ) [slot]

```



## Paramètres

<i>humiditeExterieur</i>	humidite exterieur de la ruche
<i>horodatage</i>	QString

Références [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00355 {  
00356     ui->lcdNumberHumiditeExterieur->display(humiditeExterieur);  
00357     ui->groupBoxTemperatures->setTitle("Mesures : " + horodatage);  
00358 }
```

### 9.37.3.12 afficherHumiditeInterieur

```
void RucheIhm::afficherHumiditeInterieur (  
    double humiditeInterieur,  
    QString horodatage ) [slot]
```

## Paramètres

<i>humiditeInterieur</i>	humidite interieur de la ruche
<i>horodatage</i>	QString

Références [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00310 {  
00311     ui->lcdNumberHumiditeInterieur->display(humiditeInterieur);  
00312     ui->groupBoxTemperatures->setTitle("Mesures : " + horodatage);  
00313 }
```

### 9.37.3.13 afficherLocalisationPasserelle

```
void RucheIhm::afficherLocalisationPasserelle (  
    QString longitude,  
    QString latitude ) [slot]
```

### 9.37.3.14 afficherMesuresJournalieresEnsoleillement

```
void RucheIhm::afficherMesuresJournalieresEnsoleillement ( ) [slot]
```

Références [bdd](#), [courbeEnsoleillement](#), [initialiserMarqueurAlerteEnsoleillement\(\)](#), [mesRuches](#), [BaseDeDonnees : :recuperer\(\)](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```

01046 {
01047     initialiserMarqueurAlerteEnsoleillement();
01048     QDate dateCourante = QDate::currentDate();
01049     int positionDeLaRuche = ui->comboBoxChoixRuche->currentIndex()-1;
01050     QVector<QStringList> mesuresJournalieresEnsoleillement;
01051     QString requete = "SELECT Ensoleillement, HeureMesure FROM MesuresJournalieresEnsoleillement WHERE
DateMesure = '" + dateCourante.toString("yyyy-MM-dd") + "' AND idRuche = '" +
mesRuches[positionDeLaRuche].at(0) + "' ORDER BY HeureMesure ASC";
01052     qDebug() << Q_FUNC_INFO << requete;
01053     bdd->recuperer(requete, mesuresJournalieresEnsoleillement);
01054     qDebug() << Q_FUNC_INFO << mesuresJournalieresEnsoleillement;
01055
01056     if(mesuresJournalieresEnsoleillement.size() < 1)
01057         return;
01058
01059     QVector<double> ensoleillements;
01060     QVector<double> heures;
01061
01062     for(int i=0; i < mesuresJournalieresEnsoleillement.size(); i++)
01063     {
01064         ensoleillements.push_back(mesuresJournalieresEnsoleillement.at(i).at(0).toDouble());
01065         QString heure = mesuresJournalieresEnsoleillement.at(i).at(1);
01066         QTime heureMesure = QTime::fromString(heure, "HH:mm:ss");
01067         heures.push_back(heureMesure.hour());
01068     }
01069     courbeEnsoleillement->setSamples(heures, ensoleillements);
01070     ui->qwtPlotEnsoleillement->replot();
01071 }

```

### 9.37.3.15 afficherMesuresJournalieresEnvironnement

```
void RucheIhm::afficherMesuresJournalieresEnvironnement ( ) [slot]
```

Références [bdd](#), [courbeHumiditeExterieur](#), [courbePressionAtmospherique](#), [courbeTemperatureExterieur](#), [initialiserMarqueurAlerteHumiditeExterieur\(\)](#), [initialiserMarqueurAlertePression\(\)](#), [initialiserMarqueurAlerteTemperatureExterieur\(\)](#), [mesRuches](#), [BaseDeDonnees : recuperer\(\)](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```

01007 {
01008     initialiserMarqueurAlerteTemperatureExterieur();
01009     initialiserMarqueurAlerteHumiditeExterieur();
01010     initialiserMarqueurAlertePression();
01011
01012     QDate dateCourante = QDate::currentDate();
01013     int positionDeLaRuche = ui->comboBoxChoixRuche->currentIndex()-1;
01014     QVector<QStringList> mesuresJournalieresEnvironnement;
01015     QString requete = "SELECT Temperature, Humidite, Pression, HeureMesure FROM
MesuresJournalieresEnvironnement WHERE DateMesure = '" + dateCourante.toString("yyyy-MM-dd") + "' AND idRuche = '" +
mesRuches[positionDeLaRuche].at(0) + "' ORDER BY HeureMesure ASC";
01016     qDebug() << Q_FUNC_INFO << requete;
01017     bdd->recuperer(requete, mesuresJournalieresEnvironnement);
01018     qDebug() << Q_FUNC_INFO << mesuresJournalieresEnvironnement;
01019
01020     if(mesuresJournalieresEnvironnement.size() < 1)
01021         return;
01022
01023     QVector<double> temperatures;
01024     QVector<double> humidites;
01025     QVector<double> pression;
01026     QVector<double> heures;
01027
01028     for(int i=0; i < mesuresJournalieresEnvironnement.size(); i++)
01029     {
01030         temperatures.push_back(mesuresJournalieresEnvironnement.at(i).at(0).toDouble());
01031         humidites.push_back(mesuresJournalieresEnvironnement.at(i).at(1).toDouble());
01032         pression.push_back(mesuresJournalieresEnvironnement.at(i).at(2).toDouble());
01033         QString heure = mesuresJournalieresEnvironnement.at(i).at(3);
01034         QTime heureMesure = QTime::fromString(heure, "HH:mm:ss");
01035         heures.push_back(heureMesure.hour());
01036     }
01037     courbeTemperatureExterieur->setSamples(heures, temperatures);
01038     ui->qwtPlotTemperature->replot();
01039     courbeHumiditeExterieur->setSamples(heures, humidites);
01040     ui->qwtPlotHumidite->replot();
01041     courbePressionAtmospherique->setSamples(heures, pression);
01042     ui->qwtPlotPressionAtmospherique->replot();
01043 }

```

## 9.37.3.16 afficherMesuresJournalieresRuche

```
void RucheIhm::afficherMesuresJournalieresRuche ( ) [slot]
```

Références [bdd](#), [courbeHumiditeInterieur](#), [courbeTemperatureInterieur](#), [initialiserMarqueurAlerteHumiditeInterieur\(\)](#), [InitialiserMarqueurAlerteTemperatureInterieur\(\)](#), [mesRuches](#), [BaseDeDonnees : :recuperer\(\)](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00972 {
00973     InitialiserMarqueurAlerteTemperatureInterieur();
00974     initialiserMarqueurAlerteHumiditeInterieur();
00975
00976     QDate dateCourante = QDate::currentDate();
00977     int positionDeLaRuche = ui->comboBoxChoixRuche->currentIndex()-1;
00978     QVector<QStringList> mesuresJournalieresRuche;
00979     QString requete = "SELECT Temperature, Humidite, HeureMesure FROM MesuresJournalieresRuche WHERE
DateMesure = '" + dateCourante.toString("yyyy-MM-dd") + "' AND idRuche = '" + mesRuches[positionDeLaRuche
].at(0) + "' ORDER BY HeureMesure ASC";
00980     qDebug() << Q_FUNC_INFO << requete;
00981     bdd->recuperer(requete, mesuresJournalieresRuche);
00982     qDebug() << Q_FUNC_INFO << mesuresJournalieresRuche;
00983
00984     if(mesuresJournalieresRuche.size() < 1)
00985         return;
00986
00987     QVector<double> temperatures;
00988     QVector<double> humidites;
00989     QVector<double> heures;
00990
00991     for(int i=0; i < mesuresJournalieresRuche.size(); i++)
00992     {
00993         temperatures.push_back(mesuresJournalieresRuche.at(i).at(0).toDouble());
00994         humidites.push_back(mesuresJournalieresRuche.at(i).at(1).toDouble());
00995         QString heure = mesuresJournalieresRuche.at(i).at(2);
00996         QTime heureMesure = QTime::fromString(heure, "HH:mm:ss");
00997         heures.push_back(heureMesure.hour());
00998     }
00999
01000     courbeTemperatureInterieur->setSamples(heures, temperatures);
01001     ui->qwtPlotTemperature->replot();
01002     courbeHumiditeInterieur->setSamples(heures, humidites);
01003     ui->qwtPlotHumidite->replot();
01004 }
```

## 9.37.3.17 afficherMesuresJournalieresSelectionnee

```
void RucheIhm::afficherMesuresJournalieresSelectionnee ( ) [slot]
```

Références [APP\\_TITRE](#), [bdd](#), [effacerTableau\(\)](#), [insérerMesuresEnsoleillementTableau\(\)](#), [insérerMesuresEnvironnementTableau\(\)](#), [insérerMesuresRucheTableau\(\)](#), [mesRuches](#), [BaseDeDonnees : :recuperer\(\)](#), et [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
01270 {
01271     if(ui->listeMesures->currentIndex() == 0)
01272         effacerTableau();
01273
01274     else if(ui->listeMesures->currentIndex() == 1)
01275     {
01276         effacerTableau();
01277         ui->tableauxMesures->setEditTriggers(QAbstractItemView::NoEditTriggers);
01278
01279         QDate dateCourante = QDate::currentDate();
01280         int positionDeLaRuche = ui->comboBoxChoixRuche->currentIndex()-1;
01281         QVector<QStringList> mesuresJournalieresRuche;
01282         QString requete = "SELECT Temperature, TemperatureMin, TemperatureMax, Humidite, HumiditeMin,
HumiditeMax, HeureMesure FROM MesuresJournalieresRuche WHERE DateMesure = '" + dateCourante.toString("yyyy-MM-dd")
) + "' AND idRuche = '" + mesRuches[positionDeLaRuche].at(0) + "' ORDER BY HeureMesure ASC";
01283         bdd->recuperer(requete, mesuresJournalieresRuche);
01284
01285         if(mesuresJournalieresRuche.size() == 0)
01286         {
01287             QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Aucunes
```

```

    données"));
01288         return;
01289     }
01290
01291     QStringList header; // nom des colonnes
01292     header << "Temperature" << "Temperature minimum" << "Temperature maximum" << "Humidite" << "
Humidite minimum" << "Humidite maximum" << "Heure";
01293
01294     // On fixe le nombre de colonnes
01295     ui->tableauxMesures->setColumnCount(header.size());
01296
01297     // On applique les noms des colonnes
01298     ui->tableauxMesures->setHorizontalHeaderLabels(header);
01299
01300     insererMesuresRucheTableau(mesuresJournalieresRuche);
01301
01302     // redimensionner les colonnes en fonction du contenu
01303     ui->tableauxMesures->resizeColumnsToContents();
01304     // redimensionner les lignes en fonction du contenu
01305     ui->tableauxMesures->resizeRowsToContents();
01306 }
01307 else if(ui->listeMesures->currentIndex() == 2)
01308 {
01309     effacerTableau();
01310     ui->tableauxMesures->setEditTriggers(QAbstractItemView::NoEditTriggers);
01311     QDate dateCourante = QDate::currentDate();
01312     int positionDeLaRuche = ui->comboBoxChoixRuche->currentIndex()-1;
01313     QVector<QStringList> mesuresJournalieresEnvironnement;
01314     QString requete = "SELECT Temperature, TemperatureMin, TemperatureMax, Humidite, HumiditeMin,
HumiditeMax, Pression, PressionMin, PressionMax, HeureMesure FROM MesuresJournalieresEnvironnement WHERE
DateMesure = '" + dateCourante.toString("yyyy-MM-dd") + "' AND idRuche = '" + mesRuches[positionDeLaRuche].
at(0) + "' ORDER BY HeureMesure ASC";
01315     bdd->recuperer(requete, mesuresJournalieresEnvironnement);
01316
01317     if(mesuresJournalieresEnvironnement.size() == 0)
01318     {
01319         QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Aucunes
données"));
01320         return;
01321     }
01322
01323     QStringList header; // nom des colonnes
01324     header << "Temperature" << "Temperature minimum" << "Temperature maximum" << "Humidite" << "
Humidite minimum" << "Humidite maximum" << "Pression" << "Pression minimum" << "Pression maximum" << "Heure";
01325
01326     // On fixe le nombre de colonnes
01327     ui->tableauxMesures->setColumnCount(header.size());
01328     ui->tableauxMesures->setHorizontalHeaderLabels(header);
01329
01330     // On applique les noms des colonnes
01331     insererMesuresEnvironnementTableau(mesuresJournalieresEnvironnement)
;
01332     // redimensionner les colonnes en fonction du contenu
01333     ui->tableauxMesures->resizeColumnsToContents();
01334     // redimensionner les lignes en fonction du contenu
01335     ui->tableauxMesures->resizeRowsToContents();
01336 }
01337 else if(ui->listeMesures->currentIndex() == 3)
01338 {
01339     effacerTableau();
01340     ui->tableauxMesures->setEditTriggers(QAbstractItemView::NoEditTriggers);
01341     QDate dateCourante = QDate::currentDate();
01342     int positionDeLaRuche = ui->comboBoxChoixRuche->currentIndex()-1;
01343     QVector<QStringList> mesuresJournalieresEnsoleillement;
01344     QString requete = "SELECT Ensoleillement, EnsoleillementMin, EnsoleillementMax, HeureMesure FROM
MesuresJournalieresEnsoleillement WHERE DateMesure = '" + dateCourante.toString("yyyy-MM-dd") + "' AND
idRuche = '" + mesRuches[positionDeLaRuche].at(0) + "' ORDER BY HeureMesure ASC";
01345     bdd->recuperer(requete, mesuresJournalieresEnsoleillement);
01346
01347     if(mesuresJournalieresEnsoleillement.size() == 0)
01348     {
01349         QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Aucunes
données"));
01350         return;
01351     }
01352
01353     QStringList header; // nom des colonnes
01354     header << "Ensoleillement" << "Ensoleillement minimum" << "Ensoleillement maximum" << "Heure";
01355
01356     // On fixe le nombre de colonnes
01357     ui->tableauxMesures->setColumnCount(header.size());
01358     ui->tableauxMesures->setHorizontalHeaderLabels(header);
01359
01360     insererMesuresEnsoleillementTableau(
mesuresJournalieresEnsoleillement);
01361     // redimensionner les colonnes en fonction du contenu
01362     ui->tableauxMesures->resizeColumnsToContents();
01363     // redimensionner les lignes en fonction du contenu
01364     ui->tableauxMesures->resizeRowsToContents();
01365 }
01366 }

```

## 9.37.3.18 afficherPoids

```
void RucheIhm::afficherPoids (
    double poids,
    QString horodatage ) [slot]
```

Références [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00323 {
00324     qDebug() << Q_FUNC_INFO << "poids = " << poids;
00325     ui->lcdNumberPoids->display(poids);
00326     ui->groupBoxTemperatures->setTitle("Mesures : " + horodatage);
00327 }
```

## 9.37.3.19 afficherPressionAtmospherique

```
void RucheIhm::afficherPressionAtmospherique (
    double pressionAtmospherique,
    QString horodatage ) [slot]
```

## Paramètres

<i>pressionAtmospherique</i>	pression atmospherique exterieur de la ruche
<i>horodatage</i>	QString

Références [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00370 {
00371     qDebug() << Q_FUNC_INFO << "pressionAtmospherique:" << pressionAtmospherique;
00372     ui->lcdNumberPressionAtmospherique->display(pressionAtmospherique);
00373     ui->groupBoxTemperatures->setTitle("Mesures : " + horodatage);
00374 }
```

## 9.37.3.20 afficherTemperatureExterieur

```
void RucheIhm::afficherTemperatureExterieur (
    double temperatureExterieur,
    QString horodatage ) [slot]
```

## Paramètres

<i>temperatureExterieur</i>	temperature exterieur de la ruche
<i>horodatage</i>	QString

Références [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00339 {
00340     ui->thermoTemperatureExterieur->setValue(temperatureExterieur);
00341     ui->lcdNumberTemperatureExterieur->display(temperatureExterieur);
00342     ui->groupBoxTemperatures->setTitle("Mesures : " + horodatage);
00343 }
```

### 9.37.3.21 afficherTemperatureInterieur

```
void RucheIhm::afficherTemperatureInterieur (
    double temperatureInterieur,
    QString horodatage ) [slot]
```

#### Paramètres

<i>temperatureInterieur</i>	double la température dans la ruche
<i>horodatage</i>	QString

Références [ui](#).

Référencé par [deconnecterSignaux\(\)](#), et [selectionnerRuche\(\)](#).

```
00293 {
00294     qDebug() << Q_FUNC_INFO << "temperatureInterieur=" << temperatureInterieur;
00295     ui->lcdNumberTemperatureInterieur->display(temperatureInterieur);
00296     ui->thermoTemperatureInterieur->setValue(temperatureInterieur);
00297     ui->groupBoxTemperatures->setTitle("Mesures : " + horodatage);
00298 }
```

### 9.37.3.22 creerNouvelleRuche

```
void RucheIhm::creerNouvelleRuche ( ) [private], [slot]
```

Références [mettreAJourListeRuches\(\)](#), et [nouvelleRuche](#).

Référencé par [RucheIhm\(\)](#).

```
00243 {
00244     int retour = nouvelleRuche->exec();
00245     if(retour == QDialog::Accepted)
00246     {
00247         mettreAJourListeRuches();
00248     }
00249     //QDialog::Rejected
00250 }
```

## 9.37.3.23 deconnecterSignaux()

```
void RucheIhm::deconnecterSignaux ( ) [private]
```

Références [afficherAlertesBatterie\(\)](#), [afficherAlertesEnsoleillement\(\)](#), [afficherAlertesHumiditeExterieur\(\)](#), [afficherAlertesHumiditeInterieur\(\)](#), [afficherAlertesPoids\(\)](#), [afficherAlertesPressionAtmospherique\(\)](#), [afficherAlertesTemperatureExterieur\(\)](#), [afficherAlertesTemperatureInterieur\(\)](#), [afficherBatterie\(\)](#), [afficherEnsoleillement\(\)](#), [afficherHumiditeExterieur\(\)](#), [afficherHumiditeInterieur\(\)](#), [afficherMesuresJournalieresEnsoleillement\(\)](#), [afficherMesuresJournalieresEnvironnement\(\)](#), [afficherMesuresJournalieresRuche\(\)](#), [afficherMesuresJournalieresSelectionnee\(\)](#), [afficherPoids\(\)](#), [afficherPressionAtmospherique\(\)](#), [afficherTemperatureExterieur\(\)](#), [afficherTemperatureInterieur\(\)](#), [maRuche](#), et [ui](#).

Référencé par [selectionnerRuche\(\)](#).

```
00138 {
00139     /*****SLOT AFFICHAGE MESURE*****/
00140     disconnect (maRuche, SIGNAL (nouvelleMesureTemperatureInterieurTraite (double,QString)), this, SLOT (
00141         afficherTemperatureInterieur (double,QString)));
00142     disconnect (maRuche, SIGNAL (nouvelleMesureHumiditeInterieurTraite (double,QString)), this, SLOT (
00143         afficherHumiditeInterieur (double,QString)));
00144     disconnect (maRuche, SIGNAL (nouvelleMesureTemperatureExterieurTraite (double,QString)), this, SLOT (
00145         afficherTemperatureExterieur (double,QString)));
00146     disconnect (maRuche, SIGNAL (nouvelleMesureHumiditeExterieurTraite (double,QString)), this, SLOT (
00147         afficherHumiditeExterieur (double,QString)));
00148     disconnect (maRuche, SIGNAL (nouvellePressionAtmospheriqueTraite (double,QString)), this, SLOT (
00149         afficherPressionAtmospherique (double,QString)));
00150     disconnect (maRuche, SIGNAL (nouvelleMesureEnsoleillementTraite (double,QString)), this, SLOT (
00151         afficherEnsoleillement (double,QString)));
00152     disconnect (maRuche, SIGNAL (nouvelleMesureCharge (double, QString)), this, SLOT (
00153         afficherBatterie (double, QString)));
00154     disconnect (maRuche, SIGNAL (nouvelleMesurePoids (double, QString)), this, SLOT (
00155         afficherPoids (double, QString)));
00156     /*****SLOT AFFICHAGE Alertes*****/
00157     disconnect (maRuche, SIGNAL (envoiAlertesTemperatureInterieur (
00158         SeuilsAlertes)), this, SLOT (afficherAlertesTemperatureInterieur
00159         (SeuilsAlertes)));
00160     disconnect (maRuche, SIGNAL (envoiAlertesTemperatureExterieur (
00161         SeuilsAlertes)), this, SLOT (afficherAlertesTemperatureExterieur
00162         (SeuilsAlertes)));
00163     disconnect (maRuche, SIGNAL (envoiAlertesHumiditeInterieur (SeuilsAlertes)), this, SLOT (
00164         afficherAlertesHumiditeInterieur (SeuilsAlertes)));
00165     disconnect (maRuche, SIGNAL (envoiAlertesHumiditeExterieur (SeuilsAlertes)), this, SLOT (
00166         afficherAlertesHumiditeExterieur (SeuilsAlertes)));
00167     disconnect (maRuche, SIGNAL (envoiAlertesPressionAtmospherique (
00168         SeuilsAlertes)), this, SLOT (afficherAlertesPressionAtmospherique
00169         (SeuilsAlertes)));
00170     disconnect (maRuche, SIGNAL (envoiAlertesPoids (SeuilsAlertes)), this, SLOT (
00171         afficherAlertesPoids (SeuilsAlertes)));
00172     disconnect (maRuche, SIGNAL (envoiAlertesEnsoleillement (SeuilsAlertes, double)), this,
00173         SLOT (afficherAlertesEnsoleillement (SeuilsAlertes, double)));
00174     disconnect (maRuche, SIGNAL (envoiAlertesBatterie (SeuilsAlertes, double)), this, SLOT (
00175         afficherAlertesBatterie (SeuilsAlertes, double)));
00176     /*****SLOT Courbes*****/
00177     disconnect (maRuche, SIGNAL (nouvelleMesureTemperatureInterieurTraiteParHeure()), this, SLOT (
00178         afficherMesuresJournalieresRuche ());
00179     disconnect (maRuche, SIGNAL (nouvelleMesureTemperatureExterieurTraiteParHeure()), this, SLOT (
00180         afficherMesuresJournalieresEnvironnement ());
00181     disconnect (maRuche, SIGNAL (nouvelleMesureHumiditeInterieurTraiteParHeure()), this, SLOT (
00182         afficherMesuresJournalieresRuche ());
00183     disconnect (maRuche, SIGNAL (nouvelleMesureHumiditeExterieurTraiteParHeure()), this, SLOT (
00184         afficherMesuresJournalieresEnvironnement ());
00185     disconnect (maRuche, SIGNAL (nouvelleMesurePressionAtmospheriqueTraiteParHeure()), this, SLOT (
00186         afficherMesuresJournalieresEnvironnement ());
00187     disconnect (maRuche, SIGNAL (nouvelleMesureEnsoleillementTraiteParHeure()), this, SLOT (
00188         afficherMesuresJournalieresEnsoleillement ());
00189     /*****SLOT TABLEAU*****/
00190     disconnect (ui->listeMesures, SIGNAL (currentIndexChanged (int)), this, SLOT (
00191         afficherMesuresJournalieresSelectionnee ());
00192     disconnect (maRuche, SIGNAL (nouvelleMesureTemperatureInterieurTraiteParHeure()), this, SLOT (
00193         afficherMesuresJournalieresSelectionnee ());
00194     disconnect (maRuche, SIGNAL (nouvelleMesureHumiditeInterieurTraiteParHeure()), this, SLOT (
00195         afficherMesuresJournalieresSelectionnee ());
00196     disconnect (maRuche, SIGNAL (nouvelleMesureTemperatureExterieurTraiteParHeure()), this, SLOT (
00197         afficherMesuresJournalieresSelectionnee ());
00198     disconnect (maRuche, SIGNAL (nouvelleMesureHumiditeExterieurTraiteParHeure()), this, SLOT (
00199         afficherMesuresJournalieresSelectionnee ());
00200     disconnect (maRuche, SIGNAL (nouvelleMesurePressionAtmospheriqueTraiteParHeure()), this, SLOT (
00201         afficherMesuresJournalieresSelectionnee ());
00202     disconnect (maRuche, SIGNAL (nouvelleMesureEnsoleillementTraiteParHeure()), this, SLOT (
00203         afficherMesuresJournalieresSelectionnee ());
00204 }
```

### 9.37.3.24 effacerTableau()

```
void RucheIhm::effacerTableau ( ) [private]
```

Références [ui](#).

Référencé par [afficherMesuresJournalieresSelectionnee\(\)](#), et [initialiserWidgets\(\)](#).

```
01369 {
01370     int nombreDeLignes = ui->tableauxMesures->rowCount();
01371     if(nombreDeLignes != 0)
01372     {
01373         for(int i = 0 ; i < nombreDeLignes ; i++)
01374         {
01375             ui->tableauxMesures->removeRow(0);
01376         }
01377     }
01378     QStringList header;
01379     ui->tableauxMesures->setColumnCount(header.size());
01380     ui->tableauxMesures->setHorizontalHeaderLabels(header);
01381 }
01382 }
```

### 9.37.3.25 initialiserEnsoleillementCourbe()

```
void RucheIhm::initialiserEnsoleillementCourbe ( ) [private]
```

Références [courbeEnsoleillement](#), et [ui](#).

Référencé par [initialiserCourbes\(\)](#).

```
00959 {
00960     ui->qwtPlotEnsoleillement->insertLegend(new QwtLegend(),QwtPlot::BottomLegend);
00961     courbeEnsoleillement = new QwtPlotCurve("Ensoleillement");
00962     courbeEnsoleillement->setLegendAttribute(QwtPlotCurve::LegendShowLine, true);
00963     courbeEnsoleillement->setPen(QPen(QColor(245, 176, 65)));
00964     courbeEnsoleillement->setCurveAttribute(QwtPlotCurve::Fitted); // courbe
00965     courbeEnsoleillement->setRenderHint(QwtPlotItem::RenderAntialiased);
00966     courbeEnsoleillement->setSymbol(new QwtSymbol(QwtSymbol::Cross, Qt::blue, QPen(
Qt::black), QSize(5, 5)));
00967     courbeEnsoleillement->attach(ui->qwtPlotEnsoleillement);
00968 }
```

### 9.37.3.26 initialiserHumiditeExterieurCourbe()

```
void RucheIhm::initialiserHumiditeExterieurCourbe ( ) [private]
```

Références [courbeHumiditeExterieur](#), et [ui](#).

Référencé par [initialiserCourbes\(\)](#).

```
00937 {
00938     ui->qwtPlotHumidite->insertLegend(new QwtLegend(),QwtPlot::BottomLegend);
00939     courbeHumiditeExterieur = new QwtPlotCurve("humidite exterieur");
00940     courbeHumiditeExterieur->setLegendAttribute(QwtPlotCurve::LegendShowLine, true);
00941     courbeHumiditeExterieur->setPen(QPen(Qt::green));
00942     courbeHumiditeExterieur->setCurveAttribute(QwtPlotCurve::Fitted); // courbe
00943     courbeHumiditeExterieur->setRenderHint(QwtPlotItem::RenderAntialiased);
00944     courbeHumiditeExterieur->attach(ui->qwtPlotHumidite);
00945 }
```



## 9.37.3.27 initialiserHumiditeInterieurCourbe()

```
void RucheIhm::initialiserHumiditeInterieurCourbe ( ) [private]
```

Références [courbeHumiditeInterieur](#), et [ui](#).

Référencé par [initialiserCourbes\(\)](#).

```
00927 {
00928     courbeHumiditeInterieur = new QwtPlotCurve("Humidite interieur");
00929     courbeHumiditeInterieur->setLegendAttribute(QwtPlotCurve::LegendShowLine, true);
00930     courbeHumiditeInterieur->setPen(QPen(QColor( 52, 152, 219 )));
00931     courbeHumiditeInterieur->setCurveAttribute(QwtPlotCurve::Fitted); // courbe
00932     courbeHumiditeInterieur->setRenderHint(QwtPlotItem::RenderAntialiased);
00933     courbeHumiditeInterieur->attach(ui->qwtPlotHumidite);
00934 }
```

## 9.37.3.28 initialiserMarqueurAlerteEnsoleillement()

```
void RucheIhm::initialiserMarqueurAlerteEnsoleillement ( ) [private]
```

Références [Ruche : :getAlertes\(\)](#), [Alertes : :getSeuilEnsoleillement\(\)](#), [maRuche](#), et [ui](#).

Référencé par [afficherMesuresJournalieresEnsoleillement\(\)](#).

```
01153 {
01154     QwtPlotMarker *SeuilEnsoleillement = new QwtPlotMarker();
01155     SeuilEnsoleillement->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01156     SeuilEnsoleillement->setLineStyle(QwtPlotMarker::HLine);
01157     SeuilEnsoleillement->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01158     SeuilEnsoleillement->setYValue(maRuche->getAlertes()->
getSeuilEnsoleillement());
01159     SeuilEnsoleillement->attach(ui->qwtPlotEnsoleillement);
01160
01161     if(SeuilEnsoleillement->yValue() != maRuche->getAlertes()->
getSeuilEnsoleillement())
01162     {
01163         ui->qwtPlotEnsoleillement->detachItems(QwtPlotItem::Rtti_PlotMarker,true);
01164         SeuilEnsoleillement->detach();
01165         /*SeuilEnsoleillement->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01166         SeuilEnsoleillement->setLineStyle(QwtPlotMarker::HLine);
01167         SeuilEnsoleillement->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));*/
01168         SeuilEnsoleillement->setYValue(maRuche->getAlertes()->
getSeuilEnsoleillement());
01169         SeuilEnsoleillement->attach(ui->qwtPlotEnsoleillement);
01170     }
01171
01172 }
```

## 9.37.3.29 initialiserMarqueurAlerteHumiditeExterieur()

```
void RucheIhm::initialiserMarqueurAlerteHumiditeExterieur ( ) [private]
```

Références [Ruche : :getAlertes\(\)](#), [Alertes : :getSeuilHumiditeExterieurMax\(\)](#), [Alertes : :getSeuilHumiditeExterieurMin\(\)](#), [maRuche](#), et [ui](#).

Référencé par [afficherMesuresJournalieresEnvironnement\(\)](#).

```
01124 {
01125
01126     QwtPlotMarker *seuilMinHumiditeExterieur = new QwtPlotMarker;
01127     seuilMinHumiditeExterieur->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01128     seuilMinHumiditeExterieur->setLineStyle(QwtPlotMarker::HLine);
01129     seuilMinHumiditeExterieur->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01130     seuilMinHumiditeExterieur->setYValue(maRuche->getAlertes()->
getSeuilHumiditeExterieurMin());
01131     seuilMinHumiditeExterieur->attach(ui->qwtPlotHumidite);
01132
01133     QwtPlotMarker *seuilMaxHumiditeExterieur = new QwtPlotMarker();
01134     seuilMaxHumiditeExterieur->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01135     seuilMaxHumiditeExterieur->setLineStyle(QwtPlotMarker::HLine);
01136     seuilMaxHumiditeExterieur->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01137     seuilMaxHumiditeExterieur->setYValue(maRuche->getAlertes()->
getSeuilHumiditeExterieurMax());
01138     seuilMaxHumiditeExterieur->attach(ui->qwtPlotHumidite);
01139
01140 }
```

### 9.37.3.30 initialiserMarqueurAlerteHumiditeInterieur()

```
void RucheIhm::initialiserMarqueurAlerteHumiditeInterieur ( ) [private]
```

Références [Ruche : :getAlertes\(\)](#), [Alertes : :getSeuilHumiditeInterieurMax\(\)](#), [Alertes : :getSeuilHumiditeInterieurMin\(\)](#), [maRuche](#), et [ui](#).

Référencé par [afficherMesuresJournalieresRuche\(\)](#).

```
01108 {
01109     QwtPlotMarker *seuilMinHumiditeInterieur = new QwtPlotMarker();
01110     seuilMinHumiditeInterieur->setLineStyle(QwtPlotMarker::HLine);
01111     seuilMinHumiditeInterieur->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01112     seuilMinHumiditeInterieur->setYValue(maRuche->getAlertes()->
getSeuilHumiditeInterieurMin());
01113     seuilMinHumiditeInterieur->attach(ui->qwtPlotHumidite);
01114
01115     QwtPlotMarker *seuilMaxHumiditeInterieur = new QwtPlotMarker();
01116     seuilMaxHumiditeInterieur->setLineStyle(QwtPlotMarker::HLine);
01117     seuilMaxHumiditeInterieur->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01118     seuilMaxHumiditeInterieur->setYValue(75.);
01119     seuilMaxHumiditeInterieur->setYValue(maRuche->getAlertes()->
getSeuilHumiditeInterieurMax());
01120     seuilMaxHumiditeInterieur->attach(ui->qwtPlotHumidite);
01121 }
```

### 9.37.3.31 initialiserMarqueurAlertePression()

```
void RucheIhm::initialiserMarqueurAlertePression ( ) [private]
```

Références [Ruche : :getAlertes\(\)](#), [Alertes : :getSeuilPressionAtmospherique\(\)](#), [maRuche](#), et [ui](#).

Référencé par [afficherMesuresJournalieresEnvironnement\(\)](#).

```
01143 {
01144     QwtPlotMarker *seuilPressionAtmospherique = new QwtPlotMarker();
01145     seuilPressionAtmospherique->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01146     seuilPressionAtmospherique->setLineStyle(QwtPlotMarker::HLine);
01147     seuilPressionAtmospherique->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01148     seuilPressionAtmospherique->setYValue(maRuche->getAlertes()->
getSeuilPressionAtmospherique());
01149     seuilPressionAtmospherique->attach(ui->qwtPlotPressionAtmospherique);
01150 }
```

### 9.37.3.32 initialiserMarqueurAlerteTemperatureExterieur()

```
void RucheIhm::initialiserMarqueurAlerteTemperatureExterieur ( ) [private]
```

Références [Ruche : :getAlertes\(\)](#), [Alertes : :getSeuilTemperatureExterieurMax\(\)](#), [Alertes : :getSeuilTemperatureExterieurMin\(\)](#), [maRuche](#), et [ui](#).

Référencé par [afficherMesuresJournalieresEnvironnement\(\)](#).

```
01091 {
01092     QwtPlotMarker *seuilMinTemperatureExterieur = new QwtPlotMarker();
01093     //seuilMinTemperatureExterieur->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01094     seuilMinTemperatureExterieur->setLineStyle(QwtPlotMarker::HLine);
01095     seuilMinTemperatureExterieur->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01096     seuilMinTemperatureExterieur->setYValue(maRuche->getAlertes()->
getSeuilTemperatureExterieurMin());
01097     seuilMinTemperatureExterieur->attach(ui->qwtPlotTemperature);
01098
01099     QwtPlotMarker *seuilMaxTemperatureExterieur = new QwtPlotMarker();
01100     //seuilMaxTemperatureExterieur->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01101     seuilMaxTemperatureExterieur->setLineStyle(QwtPlotMarker::HLine);
01102     seuilMaxTemperatureExterieur->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01103     seuilMaxTemperatureExterieur->setYValue(maRuche->getAlertes()->
getSeuilTemperatureExterieurMax());
01104     seuilMaxTemperatureExterieur->attach(ui->qwtPlotTemperature);
01105 }
```

## 9.37.3.33 InitialiserMarqueurAlerteTemperatureInterieur()

```
void RucheIhm::InitialiserMarqueurAlerteTemperatureInterieur ( ) [private]
```

Références [Ruche](#) : [:getAlertes\(\)](#), [Alertes](#) : [:getSeuilTemperatureInterieurMax\(\)](#), [Alertes](#) : [:getSeuilTemperatureInterieurMin\(\)](#), [maRuche](#), et [ui](#).

Référencé par [afficherMesuresJournalieresRuche\(\)](#).

```
01074 {
01075     QwtPlotMarker *seuilMinTemperatureInterieur = new QwtPlotMarker();
01076     //seuilMinTemperatureInterieur->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01077     seuilMinTemperatureInterieur->setLineStyle(QwtPlotMarker::HLine);
01078     seuilMinTemperatureInterieur->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01079     seuilMinTemperatureInterieur->setYValue(maRuche->getAlertes()->
getSeuilTemperatureInterieurMin());
01080     seuilMinTemperatureInterieur->attach(ui->qwtPlotTemperature);
01081
01082     QwtPlotMarker *seuilMaxTemperatureInterieur = new QwtPlotMarker();
01083     //seuilMaxTemperatureInterieur->setLabelAlignment(Qt::AlignRight|Qt::AlignTop);
01084     seuilMaxTemperatureInterieur->setLineStyle(QwtPlotMarker::HLine);
01085     seuilMaxTemperatureInterieur->setLinePen(QPen(Qt::red, 1.0, Qt::DashLine));
01086     seuilMaxTemperatureInterieur->setYValue(maRuche->getAlertes()->
getSeuilTemperatureInterieurMax());
01087     seuilMaxTemperatureInterieur->attach(ui->qwtPlotTemperature);
01088 }
```

## 9.37.3.34 initialiserPressionAtmospheriqueCourbe()

```
void RucheIhm::initialiserPressionAtmospheriqueCourbe ( ) [private]
```

Références [courbePressionAtmospherique](#), et [ui](#).

Référencé par [initialiserCourbes\(\)](#).

```
00948 {
00949     ui->qwtPlotPressionAtmospherique->insertLegend(new QwtLegend(),QwtPlot::BottomLegend);
00950     courbePressionAtmospherique = new QwtPlotCurve("Pression atmospherique");
00951     courbePressionAtmospherique->setLegendAttribute(QwtPlotCurve::LegendShowLine
, true);
00952     courbePressionAtmospherique->setPen(QPen(Qt::blue));
00953     courbePressionAtmospherique->setCurveAttribute(QwtPlotCurve::Fitted); //
courbe
00954     courbePressionAtmospherique->setRenderHint(QwtPlotItem::RenderAntialiased);
00955     courbePressionAtmospherique->attach(ui->qwtPlotPressionAtmospherique);
00956 }
```

## 9.37.3.35 initialiserTemperatureExterieurCourbe()

```
void RucheIhm::initialiserTemperatureExterieurCourbe ( ) [private]
```

Références [courbeTemperatureExterieur](#), et [ui](#).

Référencé par [initialiserCourbes\(\)](#).

```
00912 {
00913     qDebug() << Q_FUNC_INFO << "COURBE Temperature Exterieur";
00914     ui->qwtPlotTemperature->insertLegend(new QwtLegend(),QwtPlot::BottomLegend);
00915     courbeTemperatureExterieur = new QwtPlotCurve("Temperature Exterieur");
00916     courbeTemperatureExterieur->setLegendAttribute(QwtPlotCurve::LegendShowLine,
true);
00917     courbeTemperatureExterieur->setPen(QPen(Qt::green));
00918     courbeTemperatureExterieur->setCurveAttribute(QwtPlotCurve::Fitted); //
courbe
00919     courbeTemperatureExterieur->setRenderHint(QwtPlotItem::RenderAntialiased);
00920     //courbeTemperatureExterieur->setSymbol(new QwtSymbol(QwtSymbol::Cross, Qt::red, QPen(Qt::black),
QSize(5,5)));
00921
00922     // on trace les points x,y
00923     courbeTemperatureExterieur->attach(ui->qwtPlotTemperature);
00924 }
```

### 9.37.3.36 initialiserTemperatureInterieurCourbe()

```
void RucheIhm::initialiserTemperatureInterieurCourbe ( ) [private]
```

Références [courbeTemperatureInterieur](#), et [ui](#).

Référencé par [initialiserCourbes\(\)](#).

```
00890 {
00891     qDebug() << Q_FUNC_INFO << "COURBE Temperature Interieur";
00892
00893     // Courbes
00894     courbeTemperatureInterieur = new QwtPlotCurve("Temperature interieur");
00895
00896     // Ajout de la courbe a la legende
00897     courbeTemperatureInterieur->setLegendAttribute(QwtPlotCurve::LegendShowLine,
00898 true);
00899     //couleur de la courbe
00900     courbeTemperatureInterieur->setPen(QPen(QColor( 52, 152, 219 )));
00901     courbeTemperatureInterieur->setCurveAttribute(QwtPlotCurve::Fitted); //
00902 courbe
00903     courbeTemperatureInterieur->setRenderHint(QwtPlotItem::RenderAntialiased);
00904
00905     //type de tracé
00906     //courbeTemperatureInterieur->setSymbol(new QwtSymbol(QwtSymbol::Ellipse, Qt::yellow, QPen(Qt::black),
00907 QSize(5, 5)));
00908
00909     //attachement de la courbe a son repere
00910     courbeTemperatureInterieur->attach(ui->qwtPlotTemperature);
00911 }
```

### 9.37.3.37 initialiserThermometre()

```
void RucheIhm::initialiserThermometre ( ) const [private]
```

Références [ui](#).

Référencé par [initialiserWidgets\(\)](#).

```
00111 {
00112     ui->thermoTemperatureInterieur->setUpperBound(50);
00113     ui->thermoTemperatureInterieur->setLowerBound(-10);
00114     ui->thermoTemperatureInterieur->setScaleStepSize(5);
00115     ui->thermoTemperatureInterieur->setPipeWidth(30);
00116     ui->thermoTemperatureInterieur->alarmEnabled();
00117
00118     ui->thermoTemperatureExterieur->setUpperBound(50);
00119     ui->thermoTemperatureExterieur->setLowerBound(-10);
00120     ui->thermoTemperatureExterieur->setScaleStepSize(5);
00121     ui->thermoTemperatureExterieur->setPipeWidth(30);
00122     ui->thermoTemperatureExterieur->alarmEnabled();
00123
00124     // Initialisations Couleurs
00125     QPalette paletteCouleur = ui->thermoTemperatureInterieur->palette();
00126     paletteCouleur.setColor(QPalette::ButtonText, QColor(0,128,0));
00127     ui->thermoTemperatureInterieur->setPalette(paletteCouleur);
00128     ui->thermoTemperatureExterieur->setPalette(paletteCouleur);
00129 }
```

## 9.37.3.38 initialiserWidgets()

```
void RucheIhm::initialiserWidgets ( ) [private]
```

Références [effacerTableau\(\)](#), [initialiserThermometre\(\)](#), [reinitialiserCourbes\(\)](#), et [ui](#).

Référencé par [recevoirEtatClientConnexion\(\)](#), et [Ruchelhm\(\)](#).

```
00698 {
00699     ui->groupBoxTemperatures->setTitle("Mesures");
00700     ui->lcdNumberHumiditeInterieur->display("--");
00701     ui->lcdNumberHumiditeExterieur->display("--");
00702     ui->lcdNumberTemperatureInterieur->display("--");
00703     ui->lcdNumberTemperatureExterieur->display("--");
00704     ui->lcdNumberPressionAtmospherique->display("--");
00705     ui->lcdNumberEnsoleillement->display("--");
00706     ui->lcdNumberBatterie->display("--");
00707     ui->lcdNumberPoids->display("--");
00708     ui->labelRuche->setText("");
00709     initialiserThermometre();
00710     ui->QTabOnglet->setCurrentIndex(0); //Onglet Mesure == 0
00711
00712     ui->labelAlerteHumiditeExterieur->setText("<strong>Humidité Ext. Normale</strong>");
00713     QPixmap imageHumiditeExterieur(":/images/images/humiditeNormal.png");
00714     ui->labelAlerteImageHumiditeInterieur->setPixmap(imageHumiditeExterieur);
00715
00716     ui->labelAlerteHumiditeInterieur->setText("<strong>Humidité Int. Normale</strong>");
00717     QPixmap imageHumiditeInterieur(":/images/images/humiditeNormal.png");
00718     ui->labelAlerteImageHumiditeInterieur->setPixmap(imageHumiditeInterieur);
00719
00720     ui->labelAlerteTemperatureExterieur->setText("<strong>Température Ext. Normale</strong>");
00721     QPixmap imageTemperatureExterieur(":/images/images/temperatureNormal.png");
00722     ui->labelAlerteImageTemperature->setPixmap(imageTemperatureExterieur);
00723
00724     ui->labelAlerteTemperatureInterieur->setText("<strong>Température Int. Normale</strong>");
00725     QPixmap imageTemperatureInterieur(":/images/images/temperatureNormal.png");
00726     ui->labelAlerteImageTemperature->setPixmap(imageTemperatureInterieur);
00727
00728     ui->labelAlerteBatterie->setText("<strong>Niveau de batterie correct</strong>");
00729     QPixmap imageBatterie(":/images/images/batteryChargeMax.png");
00730     ui->labelAlerteImageBatterie->setPixmap(imageBatterie);
00731
00732     ui->labelAlertePoids->setText("<strong>Poids Normal</strong>");
00733     QPixmap imagePoids(":/images/images/poidsnormal.png");
00734     ui->labelAlerteImagePoids->setPixmap(imagePoids);
00735
00736     ui->labelAlerteEnsoleillement->setText("<strong>Ensoleillement Normal</strong>");
00737
00738     ui->labelAlertePressionAtmospherique->setText("<strong>Pression Atmospherique Normale</strong>");
00739
00740     ui->labelLocalisationPasserelle->setText("");
00741     ui->webViewPasserelle->setStyleSheet("background-color: #FFFFB9;");
00742     ui->webViewPasserelle->load(QUrl(""));
00743     //ui->webViewPasserelle->setVisible(false);
00744     ui->labelLocalisationRuche->setText("");
00745     ui->webViewRuche->setStyleSheet("background-color: #FFFFB9;");
00746     ui->webViewRuche->load(QUrl(""));
00747     //ui->webViewRuche->setVisible(false);
00748
00749     reinitialiserCourbes();
00750     ui->listeMesures->setCurrentIndex(0);
00751     effacerTableau();
00752 }
```

## 9.37.3.39 initialiserCourbes()

```
void RucheIhm::inititialiserCourbes ( ) [private]
```

Références [initialiserEnsoleillementCourbe\(\)](#), [initialiserHumiditeExterieurCourbe\(\)](#), [initialiserHumiditeInterieurCourbe\(\)](#), [initialiser←PressionAtmospheriqueCourbe\(\)](#), [initialiserTemperatureExterieurCourbe\(\)](#), [initialiserTemperatureInterieurCourbe\(\)](#), et [ui](#).

Référencé par [Ruchelhm\(\)](#).

```

00783 {
00784     //initialisation Titre
00785     ui->qwtPlotHumidite->setTitle(QString::fromUtf8("Humidité"));
00786     ui->qwtPlotPressionAtmospherique->setTitle(QString::fromUtf8("Pression atmosphérique"));
00787     ui->qwtPlotTemperature->setTitle(QString::fromUtf8("Température"));
00788     ui->qwtPlotPoids->setTitle(QString::fromUtf8("Poids"));
00789     ui->qwtPlotEnsoleillement->setTitle(QString::fromUtf8("Ensoleillement"));
00790     ui->qwtPlotCharge->setTitle(QString::fromUtf8("Charge"));
00791
00792     // une légende à droite
00793     ui->qwtPlotPoids->insertLegend(new QwtLegend(),QwtPlot::RightLegend);
00794     ui->qwtPlotCharge->insertLegend(new QwtLegend(),QwtPlot::RightLegend);
00795
00796     // cadrillage
00797     QwtPlotGrid *cadrillageHumidite = new QwtPlotGrid ;;
00798     QwtPlotGrid *cadrillagePressionAtmospherique = new QwtPlotGrid ;;
00799     QwtPlotGrid *cadrillageTemperature = new QwtPlotGrid ;;
00800     QwtPlotGrid *cadrillagePoids = new QwtPlotGrid ;;
00801     QwtPlotGrid *cadrillageEnsoleillement = new QwtPlotGrid ;;
00802     QwtPlotGrid *cadrillageCharge = new QwtPlotGrid ;
00803
00804     cadrillageHumidite->setMajorPen(QPen(Qt::black, 0, Qt::DotLine));
00805     cadrillagePressionAtmospherique->setMajorPen(QPen(Qt::black, 0, Qt::DotLine));
00806     cadrillageTemperature->setMajorPen(QPen(Qt::black, 0, Qt::DotLine));
00807     cadrillagePoids->setMajorPen(QPen(Qt::black, 0, Qt::DotLine));
00808     cadrillageEnsoleillement->setMajorPen(QPen(Qt::black, 0, Qt::DotLine));
00809     cadrillageCharge->setMajorPen(QPen(Qt::black, 0, Qt::DotLine));
00810
00811
00812     cadrillageHumidite->attach(ui->qwtPlotHumidite);
00813     cadrillagePressionAtmospherique->attach(ui->qwtPlotPressionAtmospherique);
00814     cadrillageTemperature->attach(ui->qwtPlotTemperature);
00815     cadrillagePoids->attach(ui->qwtPlotPoids);
00816     cadrillageEnsoleillement->attach(ui->qwtPlotEnsoleillement);
00817     cadrillageCharge->attach(ui->qwtPlotCharge);
00818
00819     //configuration des axes
00820     ui->qwtPlotHumidite->setAxisTitle(QwtPlot::xBottom,QString::fromUtf8("heures -->")); // nom de l'axe
x
00821     ui->qwtPlotHumidite->setAxisTitle(QwtPlot::yLeft,QString::fromUtf8("% -->")); // nom de l'Axe y
00822     ui->qwtPlotHumidite->setAxisScale(QwtPlot::xBottom, 0.0, 24.0,0); // scale de l'axe x
00823     ui->qwtPlotHumidite->setAxisScale(QwtPlot::yLeft, 0.0, 100.0); // scale de l'axe y
00824
00825     ui->qwtPlotPressionAtmospherique->setAxisTitle(QwtPlot::xBottom,QString::fromUtf8("heures -->")); //
nom de l'axe x
00826     ui->qwtPlotPressionAtmospherique->setAxisTitle(QwtPlot::yLeft,QString::fromUtf8("hPa -->")); // nom
de l'Axe y
00827     ui->qwtPlotPressionAtmospherique->setAxisScale(QwtPlot::xBottom, 0.0, 24.0, 0); // scale de l'axe x
00828     ui->qwtPlotPressionAtmospherique->setAxisScale(QwtPlot::yLeft, 0.0, 1300.0); // scale de l'axe y
00829
00830     ui->qwtPlotTemperature->setAxisTitle(QwtPlot::xBottom,QString::fromUtf8("heures -->")); // nom de
l'axe x
00831     ui->qwtPlotTemperature->setAxisTitle(QwtPlot::yLeft,QString::fromUtf8("°C -->")); // nom de l'Axe y
00832     ui->qwtPlotTemperature->setAxisScale(QwtPlot::xBottom, 0.0, 24.0, 0); // scale de l'axe x
00833     ui->qwtPlotTemperature->setAxisScale(QwtPlot::yLeft, 0.0, 55.0); // scale de l'axe y
00834
00835     ui->qwtPlotPoids->setAxisTitle(QwtPlot::xBottom,QString::fromUtf8("heures -->")); // nom de l'axe x
00836     ui->qwtPlotPoids->setAxisTitle(QwtPlot::yLeft,QString::fromUtf8("Kg -->")); // nom de l'Axe y
00837     ui->qwtPlotPoids->setAxisScale(QwtPlot::xBottom, 0.0, 24.0, 0); // scale de l'axe x
00838     ui->qwtPlotPoids->setAxisScale(QwtPlot::yLeft, 0.0, 100.0); // scale de l'axe y
00839
00840     ui->qwtPlotEnsoleillement->setAxisTitle(QwtPlot::xBottom,QString::fromUtf8("heures -->")); // nom de
l'axe x
00841     ui->qwtPlotEnsoleillement->setAxisTitle(QwtPlot::yLeft,QString::fromUtf8("W/M² -->")); // nom de
l'Axe y
00842     ui->qwtPlotEnsoleillement->setAxisScale(QwtPlot::xBottom, 0.0, 24.0, 0); // scale de l'axe x
00843     ui->qwtPlotEnsoleillement->setAxisScale(QwtPlot::yLeft, 0.0, 1500); // scale de l'axe y
00844
00845     ui->qwtPlotCharge->setAxisTitle(QwtPlot::xBottom,QString::fromUtf8("heures -->")); // nom de l'axe x
00846     ui->qwtPlotCharge->setAxisTitle(QwtPlot::yLeft,QString::fromUtf8("% -->")); // nom de l'Axe y
00847     ui->qwtPlotCharge->setAxisScale(QwtPlot::xBottom, 0.0, 24.0, 0); // scale de l'axe x
00848     ui->qwtPlotCharge->setAxisScale(QwtPlot::yLeft, 0.0, 100); // scale de l'axe y
00849
00850     initialiserTemperatureInterieurCourbe();
00851     initialiserTemperatureExterieurCourbe();
00852     initialiserHumiditeInterieurCourbe();
00853     initialiserHumiditeExterieurCourbe();
00854     initialiserPressionAtmospheriqueCourbe();
00855     initialiserEnsoleillementCourbe();
00856 }

```

### 9.37.3.40 insererMesuresEnsoleillementTableau()

```

void RucheIhm::insererMesuresEnsoleillementTableau (
    QVector< QStringList > mesuresJournalieresEnsoleillement )

```

Références [ui](#).

Référencé par [afficherMesuresJournalieresSelectionnee\(\)](#).

```

01248 {
01249     for(int i = 0 ; i < mesuresJournalieresEnsoleillement.size() ; i++ )
01250     {
01251         double ensoleillementDouble = QString(mesuresJournalieresEnsoleillement.at(i).at(0)).toDouble();
01252         double ensoleillementMinDouble = QString(mesuresJournalieresEnsoleillement.at(i).at(1)).toDouble();
01253         double ensoleillementMaxDouble = QString(mesuresJournalieresEnsoleillement.at(i).at(2)).toDouble();
01254
01255         QTableWidgetItem *itemEnsoleillement = new QTableWidgetItem(QString::number(ensoleillementDouble,
01256 'g',4));
01257         QTableWidgetItem *itemEnsoleillementMin = new QTableWidgetItem(QString::number(
01258 ensoleillementMinDouble,'g',4));
01259         QTableWidgetItem *itemEnsoleillementMax = new QTableWidgetItem(QString::number(
01260 ensoleillementMaxDouble,'g',4));
01261         QTableWidgetItem *itemHeureMesure = new QTableWidgetItem(mesuresJournalieresEnsoleillement.at(i).at
01262 (3));
01263
01264         int nombreDeLigne = ui->tableauxMesures->rowCount();
01265         ui->tableauxMesures->insertRow(nombreDeLigne);
01266         ui->tableauxMesures->setItem(nombreDeLigne, 0, itemEnsoleillement);
01267         ui->tableauxMesures->setItem(nombreDeLigne, 1, itemEnsoleillementMin);
01268         ui->tableauxMesures->setItem(nombreDeLigne, 2, itemEnsoleillementMax);
01269         ui->tableauxMesures->setItem(nombreDeLigne, 3, itemHeureMesure);
01270     }
01271 }

```

### 9.37.3.41 insérerMesuresEnvironnementTableau()

```

void RucheIhm::insérerMesuresEnvironnementTableau (
    QVector< QStringList > mesuresJournalieresEnvironnement ) [private]

```

Références [ui](#).

Référencé par [afficherMesuresJournalieresSelectionnee\(\)](#).

```

01207 {
01208     for(int i = 0; i < mesuresJournalieresEnvironnement.size(); i++)
01209     {
01210         double temperatureDouble = QString(mesuresJournalieresEnvironnement.at(i).at(0)).toDouble();
01211         double temperatureMinDouble = QString(mesuresJournalieresEnvironnement.at(i).at(1)).toDouble();
01212         double temperatureMaxDouble = QString(mesuresJournalieresEnvironnement.at(i).at(2)).toDouble();
01213         double humiditeDouble = QString(mesuresJournalieresEnvironnement.at(i).at(3)).toDouble();
01214         double humiditeMinDouble = QString(mesuresJournalieresEnvironnement.at(i).at(4)).toDouble();
01215         double humiditeMaxDouble = QString(mesuresJournalieresEnvironnement.at(i).at(5)).toDouble();
01216         double pressionDouble = QString(mesuresJournalieresEnvironnement.at(i).at(6)).toDouble();
01217         double pressionMinDouble = QString(mesuresJournalieresEnvironnement.at(i).at(7)).toDouble();
01218         double pressionMaxDouble = QString(mesuresJournalieresEnvironnement.at(i).at(8)).toDouble();
01219
01220         QTableWidgetItem *itemTemperature = new QTableWidgetItem(QString::number(temperatureDouble,'g',4));
01221         QTableWidgetItem *itemTemperatureMin = new QTableWidgetItem(QString::number(temperatureMinDouble,
01222 'g',4));
01223         QTableWidgetItem *itemTemperatureMax = new QTableWidgetItem(QString::number(temperatureMaxDouble,
01224 'g',4));
01225         QTableWidgetItem *itemHumidite = new QTableWidgetItem(QString::number(humiditeDouble,'g',4));
01226         QTableWidgetItem *itemHumiditeMin = new QTableWidgetItem(QString::number(humiditeMinDouble,'g',4));
01227         QTableWidgetItem *itemHumiditeMax = new QTableWidgetItem(QString::number(humiditeMaxDouble,'g',4));
01228         QTableWidgetItem *itemPression = new QTableWidgetItem(QString::number(pressionDouble,'g',4));
01229         QTableWidgetItem *itemPressionMin = new QTableWidgetItem(QString::number(pressionMinDouble,'g',4));
01230         QTableWidgetItem *itemPressionMax = new QTableWidgetItem(QString::number(pressionMaxDouble,'g',4));
01231         QTableWidgetItem *itemHeureMesure = new QTableWidgetItem(mesuresJournalieresEnvironnement.at(i).at(9
01232 ));
01233
01234         int nombreDeLigne = ui->tableauxMesures->rowCount();
01235         ui->tableauxMesures->insertRow(nombreDeLigne);
01236         ui->tableauxMesures->setItem(nombreDeLigne, 0, itemTemperature);
01237         ui->tableauxMesures->setItem(nombreDeLigne, 1, itemTemperatureMin);
01238         ui->tableauxMesures->setItem(nombreDeLigne, 2, itemTemperatureMax);
01239         ui->tableauxMesures->setItem(nombreDeLigne, 3, itemHumidite);
01240         ui->tableauxMesures->setItem(nombreDeLigne, 4, itemHumiditeMin);
01241         ui->tableauxMesures->setItem(nombreDeLigne, 5, itemHumiditeMax);
01242         ui->tableauxMesures->setItem(nombreDeLigne, 6, itemPression);
01243         ui->tableauxMesures->setItem(nombreDeLigne, 7, itemPressionMin);
01244         ui->tableauxMesures->setItem(nombreDeLigne, 8, itemPressionMax);
01245         ui->tableauxMesures->setItem(nombreDeLigne, 9, itemHeureMesure);
01246     }
01247 }

```

### 9.37.3.42 insererMesuresRucheTableau()

```
void RucheIhm::insererMesuresRucheTableau (
    QVector< QStringList > mesuresJournalieresRuche ) [private]
```

Références [ui](#).

Référencé par [afficherMesuresJournalieresSelectionnee\(\)](#).

```
01175 {
01176     for(int i=0; i < mesuresJournalieresRuche.size(); i++)
01177     {
01178         double temperatureDouble = QString(mesuresJournalieresRuche.at(i).at(0)).toDouble();
01179         double temperatureMinDouble = QString(mesuresJournalieresRuche.at(i).at(1)).toDouble();
01180         double temperatureMaxDouble = QString(mesuresJournalieresRuche.at(i).at(2)).toDouble();
01181         double humiditeDouble = QString(mesuresJournalieresRuche.at(i).at(3)).toDouble();
01182         double humiditeMinDouble = QString(mesuresJournalieresRuche.at(i).at(4)).toDouble();
01183         double humiditeMaxDouble = QString(mesuresJournalieresRuche.at(i).at(5)).toDouble();
01184
01185         QTableWidgetItem *itemTemperature = new QTableWidgetItem(QString::number(temperatureDouble,'g',4));
01186         QTableWidgetItem *itemTemperatureMin = new QTableWidgetItem(QString::number(temperatureMinDouble,
01187 'g',4));
01187         QTableWidgetItem *itemTemperatureMax = new QTableWidgetItem(QString::number(temperatureMaxDouble,
01188 'g',4));
01188         QTableWidgetItem *itemHumidite = new QTableWidgetItem(QString::number(humiditeDouble,'g',4));
01189         QTableWidgetItem *itemHumiditeMin = new QTableWidgetItem(QString::number(humiditeMinDouble,'g',4));
01190         QTableWidgetItem *itemHumiditeMax = new QTableWidgetItem(QString::number(humiditeMaxDouble,'g',4));
01191         QTableWidgetItem *itemHeureMesure = new QTableWidgetItem(mesuresJournalieresRuche.at(i).at(6));
01192
01193         int nombreDeLigne = ui->tableauxMesures->rowCount();
01194         ui->tableauxMesures->insertRow(nombreDeLigne);
01195         ui->tableauxMesures->setItem(nombreDeLigne, 0, itemTemperature);
01196         ui->tableauxMesures->setItem(nombreDeLigne, 1, itemTemperatureMin);
01197         ui->tableauxMesures->setItem(nombreDeLigne, 2, itemTemperatureMax);
01198         ui->tableauxMesures->setItem(nombreDeLigne, 3, itemHumidite);
01199         ui->tableauxMesures->setItem(nombreDeLigne, 4, itemHumiditeMin);
01200         ui->tableauxMesures->setItem(nombreDeLigne, 5, itemHumiditeMax);
01201         ui->tableauxMesures->setItem(nombreDeLigne, 6, itemHeureMesure);
01202     }
01203 }
01204 }
```

### 9.37.3.43 mettreAJourListeRuches

```
void RucheIhm::mettreAJourListeRuches ( ) [private], [slot]
```

Références [bdd](#), [maRuche](#), [mesRuches](#), [BaseDeDonnees](#) : [recuperer\(\)](#), et [ui](#).

Référencé par [creerNouvelleRuche\(\)](#), et [RucheIhm\(\)](#).

```
00673 {
00674     mesRuches.clear();
00675     ui->comboBoxChoixRuche->clear();
00676     QString requete = "SELECT Ruche.idRuche, Ruche.Nom, Ruche.DeviceID, TTN.idTTN, TTN.Hostname, TTN.Port,
TTN.Username, TTN.Password, TTN.ApplicationID, Ruche.Adresse, Ruche.DateMiseEnService, Ruche.Longitude,
Ruche.Latitude FROM Ruche INNER JOIN TTN ON Ruche.idTTN = TTN.idTTN ";
00677     qDebug() << Q_FUNC_INFO << requete;
00678     bdd->recuperer(requete, mesRuches);
00679     qDebug() << Q_FUNC_INFO << mesRuches;
00680     if(mesRuches.size() == 0)
00681         ui->comboBoxChoixRuche->addItem("Aucune ruche");
00682     else
00683         ui->comboBoxChoixRuche->addItem("Choix d'une ruche");
00684     for(int i=0; i<mesRuches.size(); i++)
00685     {
00686         QStringList maRuche = mesRuches.at(i);
00687         ui->comboBoxChoixRuche->addItem(maRuche.at(1));
00688     }
00689 }
```



#### 9.37.3.44 nouvelleDonnesRuche

```
void RucheIhm::nouvelleDonnesRuche (
    QString nom,
    QString prenom,
    QString email,
    QString nomTopic ) [signal]
```

#### 9.37.3.45 ouvrirReglagesAlertes

```
void RucheIhm::ouvrirReglagesAlertes ( ) [private], [slot]
```

Références [APP\\_TITRE](#), [reglagesAlertesIhm](#), et [ui](#).

Référencé par [RucheIhm\(\)](#).

```
00253 {
00254     if(ui->comboBoxChoixRuche->currentIndex() == 0)
00255     {
00256         QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Aucune ruche
selectionnée !"));
00257         return;
00258     }
00259     reglagesAlertesIhm->exec();
00260 }
```

#### 9.37.3.46 quitter

```
void RucheIhm::quitter ( ) [slot]
```

Référencé par [RucheIhm\(\)](#).

```
00098 {
00099     // Fermeture de la fenêtre
00100     close();
00101 }
```

#### 9.37.3.47 recevoirEtatClientConnexion

```
void RucheIhm::recevoirEtatClientConnexion (
    bool etat ) [slot]
```

**A faire** récupérer la longitude et la latitude depuis le format Json et afficher la position sur la carte

Références [initialiserWidgets\(\)](#), [mesRuches](#), et [ui](#).

Référencé par [selectionnerRuche\(\)](#).

```

00263 {
00264     if (etat)
00265     {
00266         ui->labelEtatConnexionRuche->setText (QString::fromUtf8("Connectée"));
00267         //ui->webViewPasserelle->setVisible(true);
00268         //ui->webViewRuche->setVisible(true);
00270         //ui->labelLocalisationPasserelle->setText ("<html><head></body><p align=\"center\"><span
style=\"font-size:18pt; font-weight:600;\">Localisation Passerelle</span></p></body></html>");
00271         //ui->webViewPasserelle->load(QUrl("https://www.openstreetmap.org/export/embed.html?bbox=" +
QString::number (mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (11).toDouble()-0.001)) + "%2C" +
QString::number (mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (12).toDouble()-0.001)) + "," +
QString::number (mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (11).toDouble()+0.001)) + "%2C" +
QString::number (mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (12).toDouble()+0.001)) + "&marker=" +
mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (12) + "," +
mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (11) + "));
00272         ui->labelLocalisationRuche->setText ("<html><head></body><p align=\"center\"><span style=\"
font-size:18pt; font-weight:600;\">Localisation Ruche</span></p></body></html>");
00273         ui->webViewRuche->load(QUrl("https://www.openstreetmap.org/export/embed.html?bbox=" +
QString::number (mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (11).toDouble()-0.001)) + "%2C" +
QString::number (mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (12).toDouble()-0.001))
+ "," + QString::number (mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (11).toDouble(
)+0.001)) + "%2C" + QString::number (mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (12
).toDouble()+0.001)) + "&marker=" + mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (12)
+ "," + mesRuches.at (ui->comboBoxChoixRuche->currentIndex()-1).at (11) + "));
00274     }
00275 }
00276 else
00277 {
00278     ui->labelEtatConnexionRuche->setText (QString::fromUtf8("Déconnectée"));
00279     initialiserWidgets();
00280 }
00281 }

```

### 9.37.3.48 reinitialiserCourbes()

```
void RucheIhm::reinitiliserCourbes ( ) [private]
```

Références [courbeEnsoleillement](#), [courbeHumiditeExterieur](#), [courbeHumiditeInterieur](#), [courbePressionAtmospherique](#), [courbeTemperatureExterieur](#), [courbeTemperatureInterieur](#), [renitialiserMarqueurs\(\)](#), et [ui](#).

Référencé par [initialiserWidgets\(\)](#).

```

00867 {
00868     //renitiliser marqueur
00869     renitialiserMarqueurs();
00870     QVector<double> temperatures;
00871     QVector<double> humidites;
00872     QVector<double> pression;
00873     QVector<double> ensoleillements;
00874     QVector<double> heures;
00875
00876     courbeTemperatureInterieur->setSamples (heures, temperatures);
00877     courbeTemperatureExterieur->setSamples (heures, temperatures);
00878     ui->qwtPlotTemperature->replot();
00879     courbeHumiditeInterieur->setSamples (heures, humidites);
00880     courbeHumiditeExterieur->setSamples (heures, humidites);
00881     ui->qwtPlotHumidite->replot();
00882     courbePressionAtmospherique->setSamples (heures, pression);
00883     ui->qwtPlotPressionAtmospherique->replot();
00884     courbeEnsoleillement->setSamples (heures, ensoleillements);
00885     ui->qwtPlotEnsoleillement->replot();
00886 }
00887 }

```

### 9.37.3.49 renitialiserMarqueurs()

```
void RucheIhm::renitialiserMarqueurs ( ) [private]
```

Références [ui](#).

Référencé par [reinitialiserCourbes\(\)](#).

```

00859 {
00860     ui->qwtPlotTemperature->detachItems( QwtPlotItem::Rtti_PlotMarker,true);
00861     ui->qwtPlotHumidite->detachItems( QwtPlotItem::Rtti_PlotMarker,true);
00862     ui->qwtPlotPressionAtmospherique->detachItems( QwtPlotItem::Rtti_PlotMarker,true);
00863     ui->qwtPlotEnsoleillement->detachItems( QwtPlotItem::Rtti_PlotMarker,true);
00864 }

```

## 9.37.3.50 selectionnerRuche

```
void RucheIhm::selectionnerRuche (
    int numeroRuche ) [private], [slot]
```

Références [afficherAlertesBatterie\(\)](#), [afficherAlertesEnsoleillement\(\)](#), [afficherAlertesHumiditeExterieur\(\)](#), [afficherAlertesHumiditeInterieur\(\)](#), [afficherAlertesPoids\(\)](#), [afficherAlertesPressionAtmospherique\(\)](#), [afficherAlertesTemperatureExterieur\(\)](#), [afficherAlertesTemperatureInterieur\(\)](#), [afficherBatterie\(\)](#), [afficherEnsoleillement\(\)](#), [afficherHumiditeExterieur\(\)](#), [afficherHumiditeInterieur\(\)](#), [afficherMesuresJournalieresEnsoleillement\(\)](#), [afficherMesuresJournalieresEnvironnement\(\)](#), [afficherMesuresJournalieresRuche\(\)](#), [afficherMesuresJournalieresSelectionnee\(\)](#), [afficherPoids\(\)](#), [afficherPressionAtmospherique\(\)](#), [afficherTemperatureExterieur\(\)](#), [afficherTemperatureInterieur\(\)](#), [deconnecterSignaux\(\)](#), [Ruche : :getAlertes\(\)](#), [maRuche](#), [mesRuches](#), [recevoirEtatClientConnexion\(\)](#), [reglagesAlertesIhm](#), [ReglagesAlertesIhm : :setAlertes\(\)](#), et [ui](#).

Référencé par [Ruchelhm\(\)](#).

```
00175 {
00176     if(mesRuches.size() == 0)
00177         return;
00178     numeroRuche -= 1;
00179     qDebug() << Q_FUNC_INFO << "numeroRuche" << numeroRuche << "nb ruches" <<
mesRuches.size();
00180     if(maRuche != 0)
00181     {
00182         deconnecterSignaux();
00183         delete maRuche;
00184         maRuche = 0;
00185     }
00186     if(maRuche == 0 && numeroRuche >= 0)
00187     {
00188         maRuche = new Ruche(mesRuches.at(numeroRuche), this);
00189         reglagesAlertesIhm->setAlertes(maRuche->
getAlertes());
00190         connect(maRuche, SIGNAL(etatClientConnexion(bool)), this, SLOT(
recevoirEtatClientConnexion(bool)));
00191         /*****SLOT AFFICHAGE MESURE*****/
00192         connect(maRuche, SIGNAL(nouvelleMesureTemperatureInterieurTraite(double,QString)), this, SLOT(
afficherTemperatureInterieur(double,QString)));
00193         connect(maRuche, SIGNAL(nouvelleMesureHumiditeInterieurTraite(double,QString)), this, SLOT(
afficherHumiditeInterieur(double,QString)));
00194         connect(maRuche, SIGNAL(nouvelleMesureTemperatureExterieurTraite(double,QString)), this, SLOT(
afficherTemperatureExterieur(double,QString)));
00195         connect(maRuche, SIGNAL(nouvelleMesureHumiditeExterieurTraite(double,QString)), this, SLOT(
afficherHumiditeExterieur(double,QString)));
00196         connect(maRuche, SIGNAL(nouvellePressionAtmospheriqueTraite(double,QString)), this, SLOT(
afficherPressionAtmospherique(double,QString)));
00197         connect(maRuche, SIGNAL(nouvelleMesureEnsoleillementTraite(double,QString)), this, SLOT(
afficherEnsoleillement(double,QString)));
00198         connect(maRuche, SIGNAL(nouvelleMesureCharge(double, QString)), this, SLOT(
afficherBatterie(double, QString)));
00199         connect(maRuche, SIGNAL(nouvelleMesurePoids(double, QString)), this, SLOT(
afficherPoids(double, QString)));
00200         /*****SLOT AFFICHAGE Alertes*****/
00201         connect(maRuche, SIGNAL(envoiAlertesTemperatureInterieur(
SeuilsAlertes)), this, SLOT(afficherAlertesTemperatureInterieur
(SeuilsAlertes)));
00202         connect(maRuche, SIGNAL(envoiAlertesTemperatureExterieur(
SeuilsAlertes)), this, SLOT(afficherAlertesTemperatureExterieur
(SeuilsAlertes)));
00203         connect(maRuche, SIGNAL(envoiAlertesHumiditeInterieur(
SeuilsAlertes)), this, SLOT(afficherAlertesHumiditeInterieur(
SeuilsAlertes)));
00204         connect(maRuche, SIGNAL(envoiAlertesHumiditeExterieur(
SeuilsAlertes)), this, SLOT(afficherAlertesHumiditeExterieur(
SeuilsAlertes)));
00205         connect(maRuche, SIGNAL(envoiAlertesPressionAtmospherique(
SeuilsAlertes)), this, SLOT(afficherAlertesPressionAtmospherique
(SeuilsAlertes)));
00206         connect(maRuche, SIGNAL(envoiAlertesPoids(SeuilsAlertes)), this, SLOT(
afficherAlertesPoids(SeuilsAlertes)));
00207         connect(maRuche, SIGNAL(envoiAlertesEnsoleillement(SeuilsAlertes,double)), this,
SLOT(afficherAlertesEnsoleillement(SeuilsAlertes, double)));
00208         connect(maRuche, SIGNAL(envoiAlertesBatterie(SeuilsAlertes, double)), this, SLOT(
afficherAlertesBatterie(SeuilsAlertes,double)));
00209         /*****SLOT Courbes*****/
00210         connect(maRuche, SIGNAL(nouvelleMesureTemperatureInterieurTraiteParHeure()), this, SLOT(
afficherMesuresJournalieresRuche()));
00211         connect(maRuche, SIGNAL(nouvelleMesureTemperatureExterieurTraiteParHeure()), this, SLOT(
afficherMesuresJournalieresEnvironnement()));
00212         connect(maRuche, SIGNAL(nouvelleMesureHumiditeInterieurTraiteParHeure()), this, SLOT(
afficherMesuresJournalieresRuche()));
```

```

00213     connect(maRuche, SIGNAL(nouvelleMesureHumiditeExterieurTraiteParHeure()), this, SLOT(
00214     afficherMesuresJournalieresEnvironnement()));
00214     connect(maRuche, SIGNAL(nouvelleMesurePressionAtmospheriqueTraiteParHeure()), this, SLOT(
00215     afficherMesuresJournalieresEnvironnement()));
00215     connect(maRuche, SIGNAL(nouvelleMesureEnsoleillementTraiteParHeure()), this, SLOT(
00216     afficherMesuresJournalieresEnsoleillement()));
00216     /*****SLOT TABLEAU*****/
00217     connect(ui->listeMesures, SIGNAL(currentIndexChanged(int)), this, SLOT(
00218     afficherMesuresJournalieresSelectionnee()));
00218     connect(maRuche, SIGNAL(nouvelleMesureTemperatureInterieurTraiteParHeure()), this, SLOT(
00219     afficherMesuresJournalieresSelectionnee()));
00219     connect(maRuche, SIGNAL(nouvelleMesureHumiditeInterieurTraiteParHeure()), this, SLOT(
00220     afficherMesuresJournalieresSelectionnee()));
00220     connect(maRuche, SIGNAL(nouvelleMesureTemperatureExterieurTraiteParHeure()), this, SLOT(
00221     afficherMesuresJournalieresSelectionnee()));
00221     connect(maRuche, SIGNAL(nouvelleMesurePressionAtmospheriqueTraiteParHeure()), this, SLOT(
00222     afficherMesuresJournalieresSelectionnee()));
00222     connect(maRuche, SIGNAL(nouvelleMesureEnsoleillementTraiteParHeure()), this, SLOT(
00223     afficherMesuresJournalieresSelectionnee()));
00224
00225     QDate date = QDate::fromString(mesRuches.at(numeroRuche).at(10), Qt::ISODate);
00226     ui->labelRuche->setText(mesRuches.at(numeroRuche).at(1) + " : " +
    mesRuches.at(numeroRuche).at(9) + " mise en service le " + date.toString("dd/MM/yyyy") + " [" +
    mesRuches.at(numeroRuche).at(12) + " " + mesRuches.at(numeroRuche).at(11) + "]);
00227
00228     //initialiserSeuilsCourbe();
00229     afficherMesuresJournalieresRuche();
00230     afficherMesuresJournalieresEnvironnement();
00231     afficherMesuresJournalieresEnsoleillement();
00232 }
00233 }

```

### 9.37.3.51 supprimerRuche

```
void RucheIhm::supprimerRuche ( ) [private], [slot]
```

Références [APP\\_TITRE](#), [bdd](#), [BaseDeDonnees : :executer\(\)](#), [mesRuches](#), et [ui](#).

Référencé par [RuchelIhm\(\)](#).

```

00755 {
00756
00757
00758     if(ui->comboBoxChoixRuche->currentIndex() == 0)
00759     {
00760         QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Suppression
    impossible ! Aucune ruche sélectionnée !"));
00761         return;
00762     }
00763     QMessageBox::StandardButton reply = QMessageBox::warning(0, "Voulez vous supprimer cette ruche ? ", "
    Toutes les mesures et données lui correspondant seront supprimé !", QMessageBox::Ok |QMessageBox::Cancel);
00764
00765     if (reply == QMessageBox::Ok)
00766     {
00767         int positionDeLaRuche = ui->comboBoxChoixRuche->currentIndex()-1;
00768         QString requete = "DELETE FROM Ruche WHERE Ruche.DeviceID =' " +
    mesRuches[positionDeLaRuche].at(2) + "'";
00769         bool etatRequete = bdd->executer(requete);
00770         if(etatRequete == true)
00771         {
00772             ui->comboBoxChoixRuche->removeItem(ui->comboBoxChoixRuche->currentIndex());
00773             ui->comboBoxChoixRuche->setCurrentIndex(0);
00774         }
00775         else
00776         {
00777             QMessageBox::critical(0, QString::fromUtf8(APP_TITRE), QString::fromUtf8("Impossible
    de supprimer la ruche"));
00778         }
00779     }
00780 }

```

### 9.37.4 Documentation des données membres

#### 9.37.4.1 bdd

```
BaseDeDonnees* RucheIhm::bdd [private]
```

Référencé par [afficherAlertesBatterie\(\)](#), [afficherAlertesEnsoleillement\(\)](#), [afficherAlertesHumiditeExterieur\(\)](#), [afficherAlertesHumiditeInterieur\(\)](#), [afficherAlertesPoids\(\)](#), [afficherAlertesPressionAtmospherique\(\)](#), [afficherAlertesTemperatureExterieur\(\)](#), [afficherAlertesTemperatureInterieur\(\)](#), [afficherMesuresJournalieresEnsoleillement\(\)](#), [afficherMesuresJournalieresEnvironnement\(\)](#), [afficherMesuresJournalieresRuche\(\)](#), [afficherMesuresJournalieresSelectionnee\(\)](#), [mettreAJourListeRuches\(\)](#), [RucheIhm\(\)](#), et [supprimerRuche\(\)](#).

#### 9.37.4.2 courbeEnsoleillement

```
QwtPlotCurve* RucheIhm::courbeEnsoleillement [private]
```

Référencé par [afficherMesuresJournalieresEnsoleillement\(\)](#), [initialiserEnsoleillementCourbe\(\)](#), et [reinitialiserCourbes\(\)](#).

#### 9.37.4.3 courbeHumiditeExterieur

```
QwtPlotCurve* RucheIhm::courbeHumiditeExterieur [private]
```

Référencé par [afficherMesuresJournalieresEnvironnement\(\)](#), [initialiserHumiditeExterieurCourbe\(\)](#), et [reinitialiserCourbes\(\)](#).

#### 9.37.4.4 courbeHumiditeInterieur

```
QwtPlotCurve* RucheIhm::courbeHumiditeInterieur [private]
```

Référencé par [afficherMesuresJournalieresRuche\(\)](#), [initialiserHumiditeInterieurCourbe\(\)](#), et [reinitialiserCourbes\(\)](#).

#### 9.37.4.5 courbePressionAtmospherique

```
QwtPlotCurve* RucheIhm::courbePressionAtmospherique [private]
```

Référencé par [afficherMesuresJournalieresEnvironnement\(\)](#), [initialiserPressionAtmospheriqueCourbe\(\)](#), et [reinitialiserCourbes\(\)](#).

#### 9.37.4.6 courbeTemperatureExterieur

```
QwtPlotCurve* RucheIhm::courbeTemperatureExterieur [private]
```

Référencé par [afficherMesuresJournalieresEnvironnement\(\)](#), [initialiserTemperatureExterieurCourbe\(\)](#), et [reinitialiserCourbes\(\)](#).

#### 9.37.4.7 courbeTemperatureInterieur

```
QwtPlotCurve* RucheIhm::courbeTemperatureInterieur [private]
```

Référencé par [afficherMesuresJournalieresRuche\(\)](#), [initialiserTemperatureInterieurCourbe\(\)](#), et [reinitialiserCourbes\(\)](#).

#### 9.37.4.8 maRuche

`Ruche*` `RucheIhm::maRuche` [private]

Référencé par `afficherAlertesBatterie()`, `afficherAlertesEnsoleillement()`, `afficherAlertesHumiditeExterieur()`, `afficherAlertesHumiditeInterieur()`, `afficherAlertesPoids()`, `afficherAlertesPressionAtmospherique()`, `afficherAlertesTemperatureExterieur()`, `afficherAlertesTemperatureInterieur()`, `deconnecterSignaux()`, `initialiserMarqueurAlerteEnsoleillement()`, `initialiserMarqueurAlerteHumiditeExterieur()`, `initialiserMarqueurAlerteHumiditeInterieur()`, `initialiserMarqueurAlertePression()`, `initialiserMarqueurAlerteTemperatureExterieur()`, `InitialiserMarqueurAlerteTemperatureInterieur()`, `mettreAJourListeRuches()`, et `selectionnerRuche()`.

#### 9.37.4.9 mesRuches

`QVector<QStringList>` `RucheIhm::mesRuches` [private]

Référencé par `afficherMesuresJournalieresEnsoleillement()`, `afficherMesuresJournalieresEnvironnement()`, `afficherMesuresJournalieresRuche()`, `afficherMesuresJournalieresSelectionnee()`, `mettreAJourListeRuches()`, `recevoirEtatClientConnexion()`, `selectionnerRuche()`, et `supprimerRuche()`.

#### 9.37.4.10 nouvelleRuche

`NouvelleRucheIhm*` `RucheIhm::nouvelleRuche` [private]

Référencé par `creerNouvelleRuche()`, et `RuchelIhm()`.

#### 9.37.4.11 reglagesAlertesIhm

`ReglagesAlertesIhm*` `RucheIhm::reglagesAlertesIhm` [private]

Référencé par `ouvrirReglagesAlertes()`, `RuchelIhm()`, et `selectionnerRuche()`.

#### 9.37.4.12 ui

`Ui::RucheIhm*` `RucheIhm::ui` [private]

Référencé par `afficherAlertesBatterie()`, `afficherAlertesEnsoleillement()`, `afficherAlertesHumiditeExterieur()`, `afficherAlertesHumiditeInterieur()`, `afficherAlertesPoids()`, `afficherAlertesPressionAtmospherique()`, `afficherAlertesTemperatureExterieur()`, `afficherAlertesTemperatureInterieur()`, `afficherBatterie()`, `afficherEnsoleillement()`, `afficherHumiditeExterieur()`, `afficherHumiditeInterieur()`, `afficherMesuresJournalieresEnsoleillement()`, `afficherMesuresJournalieresEnvironnement()`, `afficherMesuresJournalieresRuche()`, `afficherMesuresJournalieresSelectionnee()`, `afficherPoids()`, `afficherPressionAtmospherique()`, `afficherTemperatureExterieur()`, `afficherTemperatureInterieur()`, `deconnecterSignaux()`, `effacerTableau()`, `initialiserEnsoleillementCourbe()`, `initialiserHumiditeExterieurCourbe()`, `initialiserHumiditeInterieurCourbe()`, `initialiserMarqueurAlerteEnsoleillement()`, `initialiserMarqueurAlerteHumiditeExterieur()`, `initialiserMarqueurAlerteHumiditeInterieur()`, `initialiserMarqueurAlertePression()`, `initialiserMarqueurAlerteTemperatureExterieur()`, `InitialiserMarqueurAlerteTemperatureInterieur()`, `initialiserPressionAtmospheriqueCourbe()`, `initialiserTemperatureExterieurCourbe()`, `initialiserTemperatureInterieurCourbe()`, `initialiserThermometre()`, `initialiserWidgets()`, `inititialiserCourbes()`, `insererMesuresEnsoleillementTableau()`, `insererMesuresEnvironnementTableau()`, `insererMesuresRucheTableau()`, `mettreAJourListeRuches()`, `ouvrirReglagesAlertes()`, `recevoirEtatClientConnexion()`, `reinitialiserCourbes()`, `renitialiserMarqueurs()`, `RuchelIhm()`, `selectionnerRuche()`, `supprimerRuche()`, et `~RuchelIhm()`.

La documentation de cette classe a été générée à partir des fichiers suivants :

- `ruchelIhm.h`
- `ruchelIhm.cpp`

## 9.38 Référence de la classe fr.campus.laurainc.honeybee.TimeAsXAxisLabelFormatter

Graphe de collaboration de fr.campus.laurainc.honeybee.TimeAsXAxisLabelFormatter :

fr.campus.laurainc.honeybee. TimeAsXAxisLabelFormatter
# mFormat
+ TimeAsXAxisLabelFormatter() + formatLabel()

## Fonctions membres publiques

- [TimeAsXAxisLabelFormatter](#) (String format)
- String [formatLabel](#) (double value, boolean isValueX)

## Attributs protégés

- final String [mFormat](#)

## 9.38.1 Documentation des constructeurs et destructeur

## 9.38.1.1 TimeAsXAxisLabelFormatter()

```
fr.campus.laurainc.honeybee.TimeAsXAxisLabelFormatter.TimeAsXAxisLabelFormatter (
    String format )

00016                                     {
00017     mFormat = format;
00018 }
```

## 9.38.2 Documentation des fonctions membres

## 9.38.2.1 formatLabel()

```
String fr.campus.laurainc.honeybee.TimeAsXAxisLabelFormatter.formatLabel (
    double value,
    boolean isValueX )

00021                                     {
00022     if (isValueX) {
00023         Date d = new Date(0, 0, 0, (int)value, 0);
00024         SimpleDateFormat dateFormat = new SimpleDateFormat(mFormat);
00025         return dateFormat.format(d);
00026     } else {
00027         return super.formatLabel(value, isValueX);
00028     }
00029 }
```

### 9.38.3 Documentation des données membres

#### 9.38.3.1 mFormat

```
final String fr.campus.laurainc.honeybee.TimeAsXAxisLabelFormatter.mFormat [protected]
```

La documentation de cette classe a été générée à partir du fichier suivant :

— [TimeAsXAxisLabelFormatter.java](#)

## 10 Documentation des fichiers

### 10.1 Référence du fichier alertes.cpp

Définition de la classe [Alertes](#).

```
#include "alertes.h"
#include "infosHumidite.h"
#include "infosPressionAtmospherique.h"
#include "infosTemperature.h"
#include "infosEnsoleillement.h"
#include "infosPoids.h"
#include "infosBatterie.h"
#include "parametres.h"
#include "reglagesAlertesIhm.h"
#include "baseDeDonnees.h"
#include "simple-mail/SmtP Mime"
#include "ruche.h"
#include <QDebug>
```

#### 10.1.1 Description détaillée

##### Auteur

Florentin Mellah, Enzo Rossi

##### Version

1.1

### 10.2 Référence du fichier alertes.h

La classe des alertes.

```
#include <QObject>
#include "parametres.h"
```



**Classes**

- class [Alertes](#)  
*La classe des alertes.*

**10.2.1 Description détaillée****Auteur**

Florentin Mellah, Enzo Rossi

**Version**

1.1

**10.3 Référence du fichier alertesActivity.java****Classes**

- class [fr.campus.laurainc.honeybee.alertesActivity](#)

**Paquetages**

- package [fr.campus.laurainc.honeybee](#)

**10.4 Référence du fichier baseDeDonnees.cpp**

Définition de la classe [BaseDeDonnees](#).

```
#include "baseDeDonnees.h"
#include "parametres.h"
#include <QDebug>
#include <QMessageBox>
```

**10.4.1 Description détaillée****Auteur**

Thierry Vaira

**Version**

0.2

**10.5 Référence du fichier baseDeDonnees.h**

Déclaration de la classe [BaseDeDonnees](#).

```
#include <QObject>
#include <QtSql/QtSql>
#include <QSqlDatabase>
#include <QMutex>
#include <QString>
#include "parametres.h"
```

## Classes

- class [BaseDeDonnees](#)

### 10.5.1 Description détaillée

## Auteur

Thierry VAIRA

## Version

0.2

## 10.6 Référence du fichier BaseDeDonnees.java

## Classes

- class [fr.campus.laurainc.honeybee.BaseDeDonnees](#)  
*Gestion d'une base de données MySQL (Singleton)*
- class [fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySQL](#)  
*Classe permettant de se connecter à MySQL en arrière-plan.*

## Paquetages

- package [fr.campus.laurainc.honeybee](#)

## 10.7 Référence du fichier CarteActivity.java

## Classes

- class [fr.campus.laurainc.honeybee.CarteActivity](#)

## Paquetages

- package [fr.campus.laurainc.honeybee](#)

## 10.8 Référence du fichier Changelog.md

## 10.9 Référence du fichier ClientMQTT.java

## Classes

- class [fr.campus.laurainc.honeybee.ClientMQTT](#)

## Paquetages

- package [fr.campus.laurainc.honeybee](#)

## 10.10 Référence du fichier communication.cpp

Définition de la classe [Communication](#).

```
#include "communication.h"  
#include <QMessageBox>  
#include <QDebug>
```

### 10.10.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.11 Référence du fichier communication.h

Déclaration de la classe [Communication](#).

```
#include <QtMqtt/QtMqtt>  
#include <QtMqtt/QtMqttClient>  
#include "parametres.h"
```

#### Classes

— class [Communication](#)  
*La classe [Communication](#).*

### 10.11.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.12 Référence du fichier DashboardActivity.java

#### Classes

— class [fr.campus.laurainc.honeybee.DashboardActivity](#)

#### Paquetages

— package [fr.campus.laurainc.honeybee](#)

### 10.13 Référence du fichier DetailsRucheActivity.java

#### Classes

- class [fr.campus.laurainc.honeybee.DetailsRucheActivity](#)

#### Paquetages

- package [fr.campus.laurainc.honeybee](#)

### 10.14 Référence du fichier ExampleInstrumentedTest.java

#### Classes

- class [fr.campus.laurainc.honeybee.ExampleInstrumentedTest](#)

#### Paquetages

- package [fr.campus.laurainc.honeybee](#)

### 10.15 Référence du fichier ExampleUnitTest.java

#### Classes

- class [fr.campus.laurainc.honeybee.ExampleUnitTest](#)

#### Paquetages

- package [fr.campus.laurainc.honeybee](#)

### 10.16 Référence du fichier GraphActivity.java

#### Classes

- class [fr.campus.laurainc.honeybee.GraphActivity](#)

#### Paquetages

- package [fr.campus.laurainc.honeybee](#)

### 10.17 Référence du fichier homeActivity.java

#### Classes

- class [fr.campus.laurainc.honeybee.homeActivity](#)

## Paquetages

— package [fr.campus.laurainc.honeybee](#)

## 10.18 Référence du fichier HoneyBee.java

## Classes

— class [fr.campus.laurainc.honeybee.HoneyBee](#)  
*Paramètres globaux de l'application.*

## Paquetages

— package [fr.campus.laurainc.honeybee](#)

## 10.19 Référence du fichier infosBatterie.cpp

Définition de la classe [InfosBatterie](#).

```
#include "infosBatterie.h"  
#include <QDebug>
```

## 10.19.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 10.20 Référence du fichier infosBatterie.h

Déclaration de la classe [InfosBatterie](#).

```
#include <QString>  
#include <QtCore/QObject>
```

## Classes

— class [InfosBatterie](#)  
*Déclaration de la classe [InfosBatterie](#).*

### 10.20.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.21 Référence du fichier infosEnsoleillement.cpp

Définition de la classe [InfosEnsoleillement](#).

```
#include "infosEnsoleillement.h"
#include <QTime>
#include <QDebug>
```

### 10.21.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.22 Référence du fichier infosEnsoleillement.h

Déclaration de la classe [InfosEnsoleillement](#).

```
#include <QString>
#include <QtCore/QObject>
#include <QVector>
```

#### Classes

— class [InfosEnsoleillement](#)  
*La classe [InfosEnsoleillement](#).*

### 10.22.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.23 Référence du fichier infosHumidite.cpp

Définition de la classe [InfosHumidite](#).

```
#include "infosHumidite.h"  
#include <QTime>  
#include <QDebug>
```

### 10.23.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.24 Référence du fichier infosHumidite.h

Déclaration de la classe [InfosHumidite](#).

```
#include <QString>  
#include <QtCore/QObject>  
#include <QVector>
```

#### Classes

— class [InfosHumidite](#)  
*La classe [InfosHumidite](#).*

### 10.24.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.25 Référence du fichier infosPoids.cpp

Défénition de la classe [InfosPoids](#).

```
#include "infosPoids.h"  
#include <QDebug>
```

### 10.25.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

### 10.26 Référence du fichier infosPoids.h

Déclaration de la classe [InfosPoids](#).

```
#include <QString>
#include <QtCore/QObject>
```

#### Classes

— class [InfosPoids](#)  
*Déclaration de la classe [InfosPoids](#).*

### 10.26.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

### 10.27 Référence du fichier infosPressionAtmospherique.cpp

Définition de la classe [InfosPressionAtmospherique](#).

```
#include "infosPressionAtmospherique.h"
#include <QDebug>
#include <QTime>
```

### 10.27.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1



## 10.28 Référence du fichier infosPressionAtmospherique.h

Déclaration de la classe [InfosPressionAtmospherique](#).

```
#include <QString>
#include <QtCore/QObject>
#include <QVector>
```

### Classes

— class [InfosPressionAtmospherique](#)  
*La classe [InfosPressionAtmospherique](#).*

#### 10.28.1 Description détaillée

##### Auteur

Florentin Mellah, Enzo Rossi

##### Version

1.1

## 10.29 Référence du fichier infosTemperature.cpp

Définition de la classe [InfosTemperature](#).

```
#include "infosTemperature.h"
#include <QDebug>
#include <QTime>
```

#### 10.29.1 Description détaillée

##### Auteur

Florentin Mellah, Enzo Rossi

##### Version

1.1

## 10.30 Référence du fichier infosTemperature.h

Déclaration de la classe [InfosTemperature](#).

```
#include <QString>
#include <QtCore/QObject>
#include <QVector>
```

**Classes**

- class [InfosTemperature](#)  
*La classe [InfosTemperature](#).*

**10.30.1 Description détaillée****Auteur**

Florentin Mellah, Enzo Rossi

**Version**

1.1

**10.31 Référence du fichier INSTALL.md****10.32 Référence du fichier main.cpp**

Programme principal.

```
#include "rucheIhm.h"
#include <QApplication>
#include <QString>
#include <QTranslator>
#include <QLocale>
#include <QLibraryInfo>
#include <baseDeDonnees.h>
```

**Fonctions**

- int [main](#) (int argc, char \*argv[])

**10.32.1 Description détaillée**

Crée et affiche la fenêtre principale de l'application

**Auteur**

Florentin Mellah, Enzo Rossi

**Version**

1.1

**10.32.2 Documentation des fonctions****10.32.2.1 main()**

```
main (
    int argc,
    char * argv[] )
```

## Paramètres

<i>argc</i>	
<i>argv[]</i>	

## Renvoie

int

```

00029 {
00030     QApplication a(argc, argv);
00031     QString locale = QLocale::system().name().section('_', 0, 0);
00032     QTranslator translator;
00033     translator.load(QString("qt_") + locale, QLibraryInfo::location(QLibraryInfo::TranslationsPath));
00034     a.installTranslator(&translator);
00035
00036     RucheIhm w;
00037     w.show();
00038
00039     return a.exec();
00040 }

```

## 10.33 Référence du fichier MainActivity.java

## Classes

- class [fr.campus.laurainc.honeybee.MainActivity](#)  
*Activité principale de l'application (Thread UI)*

## Paquetages

- package [fr.campus.laurainc.honeybee](#)

## 10.34 Référence du fichier NouvelleRucheActivity.java

## Classes

- class [fr.campus.laurainc.honeybee.NouvelleRucheActivity](#)

## Paquetages

- package [fr.campus.laurainc.honeybee](#)

## 10.35 Référence du fichier nouvelleRucheIhm.cpp

La classe [NouvelleRucheIhm](#).

```

#include "nouvelleRucheIhm.h"
#include "ui_nouvelleRucheIhm.h"
#include "baseDeDonnees.h"
#include <QMessageBox>

```

### 10.35.1 Description détaillée

Définition de la classe [Ruchelhm](#).

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

### 10.36 Référence du fichier nouvelleRuchelhm.h

```
#include <QDialog>
```

#### Classes

- class [NouvelleRuchelhm](#)  
Déclaration de la classe [NouvelleRuchelhm](#).

#### Espaces de nommage

- [Ui](#)

### 10.37 Référence du fichier parametres.h

Paramètres généraux de l'application.

#### Macros

```
— #define APP_TITRE "Projet Ruche 2019"
— #define BDD_SERVEUR "192.168.52.119"
— #define BDD_USERNAME "fmellah"
— #define BDD_PASSWORD "password"
— #define BDD_NOMBASE "ruches"
— #define TTN_SERVEUR "eu.thethings.network"
— #define TTN_PORT 1883
— #define TTN_USERNAME "mes_ruches"
— #define TTN_PASSWORD "ttn-account-v2.vC-aqMRnLLzGkNjODWgy81kLqzxBPAT8_mE-L7U2C_w"
— #define TTN_TOPIC "mes_ruches/devices/ruche_1/up"
— #define TTN_EMAIL "florentinmellah@gmail.com"
— #define USER_GMAIL QLatin1String("bee.honey.bts@gmail.com")
— #define PASSWORD_GMAIL QLatin1String("ruches123")
— #define TEMPERATURE_INTERIEUR_SEUIL_MAX 35.0
— #define TEMPERATURE_INTERIEUR_SEUIL_MIN 25.
— #define HUMIDITE_INTERIEUR_SEUIL_MAX 30.
— #define HUMIDITE_INTERIEUR_SEUIL_MIN 20.
— #define TEMPERATURE_EXTERIEUR_SEUIL_MAX 35.
— #define TEMPERATURE_EXTERIEUR_SEUIL_MIN 5.
— #define HUMIDITE_EXTERIEUR_SEUIL_MAX 35.
— #define HUMIDITE_EXTERIEUR_SEUIL_MIN 20.
— #define PRESSION_ATMOSPHERIQUE_SEUIL_MIN 1000.
— #define PRESSION_ATMOSPHERIQUE_SEUIL_MAX 1200.
— #define POIDS_SEUIL_MAX 100.
— #define POIDS_SEUIL_MIN 35.
— #define ENSOLEILLEMENT_SEUIL_MAX 1000.
— #define ENSOLEILLEMENT_SEUIL_MIN 10.
— #define BATTERIE_SEUIL_MIN 25.
```

## Énumérations

```

— enum SeuilsAlertes { tropHaut = 0, tropBas = 1, bon = 2 }
— enum TypeAlertes {
    alerteTemperatureInterieur = 0, alerteTemperatureExterieur = 1, alerteHumiditeInterieur = 2, alerteHumiditeExterieur = 3,
    alertePressionAtmospherique = 4, alertePoids = 5, alerteEnsoleillement = 6, alerteBatterie = 7,
    toutesLesAlertes = 8 }
— enum PortsTTN {
    portInconnu = 0, portMesureEnergie = 1, portMesurePoids, portMesureRuche,
    portMesureEnvironnement, portMesureEnsoleillement, portVol, nbPortsTTN }

```

## 10.37.1 Description détaillée

## Auteur

Florentin Mellah, Enzo Rossi

## Version

1.1

## 10.37.2 Documentation des macros

## 10.37.2.1 APP\_TITRE

```
#define APP_TITRE "Projet Ruche 2019"
```

Référencé par `Ruchelhm : :afficherMesuresJournalieresSelectionnee()`, `BaseDeDonnees : :connecter()`, `Communication : :connecte←TTN()`, `Ruchelhm : :ouvrirReglagesAlertes()`, `NouvelleRuchelhm : :recevoirDonneeAjoutRuche()`, et `Ruchelhm : :supprimerRuche()`.

## 10.37.2.2 BATTERIE\_SEUIL\_MIN

```
#define BATTERIE_SEUIL_MIN 25.
```

## 10.37.2.3 BDD\_NOMBASE

```
#define BDD_NOMBASE "ruches"
```

Référencé par `Alertes : :Alertes()`, `NouvelleRuchelhm : :NouvelleRuchelhm()`, `ReglagesAlertesIhm : :ReglagesAlertesIhm()`, `Ruche←: :Ruche()`, et `Ruchelhm : :Ruchelhm()`.

## 10.37.2.4 BDD\_PASSWORD

```
#define BDD_PASSWORD "password"
```

Référencé par `Alertes : :Alertes()`, `NouvelleRuchelhm : :NouvelleRuchelhm()`, `ReglagesAlertesIhm : :ReglagesAlertesIhm()`, `Ruche←: :Ruche()`, et `Ruchelhm : :Ruchelhm()`.

#### 10.37.2.5 BDD\_SERVEUR

```
#define BDD_SERVEUR "192.168.52.119"
```

Référencé par [Alertes : :Alertes\(\)](#), [NouvelleRucheIhm : :NouvelleRucheIhm\(\)](#), [ReglagesAlertesIhm : :ReglagesAlertesIhm\(\)](#), [Ruche↔ : :Ruche\(\)](#), et [RucheIhm : :RucheIhm\(\)](#).

#### 10.37.2.6 BDD\_USERNAME

```
#define BDD_USERNAME "fmellah"
```

Référencé par [Alertes : :Alertes\(\)](#), [NouvelleRucheIhm : :NouvelleRucheIhm\(\)](#), [ReglagesAlertesIhm : :ReglagesAlertesIhm\(\)](#), [Ruche↔ : :Ruche\(\)](#), et [RucheIhm : :RucheIhm\(\)](#).

#### 10.37.2.7 ENSOLEILLEMENT\_SEUIL\_MAX

```
#define ENSOLEILLEMENT_SEUIL_MAX 1000.
```

#### 10.37.2.8 ENSOLEILLEMENT\_SEUIL\_MIN

```
#define ENSOLEILLEMENT_SEUIL_MIN 10.
```

#### 10.37.2.9 HUMIDITE\_EXTERIEUR\_SEUIL\_MAX

```
#define HUMIDITE_EXTERIEUR_SEUIL_MAX 35.
```

#### 10.37.2.10 HUMIDITE\_EXTERIEUR\_SEUIL\_MIN

```
#define HUMIDITE_EXTERIEUR_SEUIL_MIN 20.
```

#### 10.37.2.11 HUMIDITE\_INTERIEUR\_SEUIL\_MAX

```
#define HUMIDITE_INTERIEUR_SEUIL_MAX 30.
```

#### 10.37.2.12 HUMIDITE\_INTERIEUR\_SEUIL\_MIN

```
#define HUMIDITE_INTERIEUR_SEUIL_MIN 20.
```

**10.37.2.13 PASSWORD\_GMAIL**

```
#define PASSWORD_GMAIL QLatin1String("ruches123")
```

Référencé par [Alertes](#) : `:envoyerMailAlerte()`.

**10.37.2.14 POIDS\_SEUIL\_MAX**

```
#define POIDS_SEUIL_MAX 100.
```

**10.37.2.15 POIDS\_SEUIL\_MIN**

```
#define POIDS_SEUIL_MIN 35.
```

**10.37.2.16 PRESSION\_ATMOSPHERIQUE\_SEUIL\_MAX**

```
#define PRESSION_ATMOSPHERIQUE_SEUIL_MAX 1200.
```

Référencé par [ReglagesAlertesIhm](#) : `:recevoirReglagesAlertes()`.

**10.37.2.17 PRESSION\_ATMOSPHERIQUE\_SEUIL\_MIN**

```
#define PRESSION_ATMOSPHERIQUE_SEUIL_MIN 1000.
```

**10.37.2.18 TEMPERATURE\_EXTERIEUR\_SEUIL\_MAX**

```
#define TEMPERATURE_EXTERIEUR_SEUIL_MAX 35.
```

**10.37.2.19 TEMPERATURE\_EXTERIEUR\_SEUIL\_MIN**

```
#define TEMPERATURE_EXTERIEUR_SEUIL_MIN 5.
```

**10.37.2.20 TEMPERATURE\_INTERIEUR\_SEUIL\_MAX**

```
#define TEMPERATURE_INTERIEUR_SEUIL_MAX 35.0
```

**10.37.2.21 TEMPERATURE\_INTERIEUR\_SEUIL\_MIN**

```
#define TEMPERATURE_INTERIEUR_SEUIL_MIN 25.
```

**10.37.2.22 TTN\_EMAIL**

```
#define TTN_EMAIL "florentinmellah@gmail.com"
```

Référencé par [Alertes](#) : `:envoyerMailAlerte()`.

**10.37.2.23 TTN\_PASSWORD**

```
#define TTN_PASSWORD "ttn-account-v2.vC-aqMRnLLzGkNjODWgy81kLqzxBPAT8_mE-L7U2C_w"
```

**10.37.2.24 TTN\_PORT**

```
#define TTN_PORT 1883
```

**10.37.2.25 TTN\_SERVEUR**

```
#define TTN_SERVEUR "eu.thethings.network"
```

**10.37.2.26 TTN\_TOPIC**

```
#define TTN_TOPIC "mes_ruches/devices/ruche_1/up"
```

**10.37.2.27 TTN\_USERNAME**

```
#define TTN_USERNAME "mes_ruches"
```

**10.37.2.28 USER\_GMAIL**

```
#define USER_GMAIL QLatin1String("bee.honey.bts@gmail.com")
```

Référencé par [Alertes](#) : `:envoyerMailAlerte()`.

**10.37.3 Documentation du type de l'énumération****10.37.3.1 PortsTTN**

```
enum PortsTTN
```



## Valeurs énumérées

portInconnu	
portMesureEnergie	
portMesurePoids	
portMesureRuche	
portMesureEnvironnement	
portMesureEnsoleillement	
portVol	
nbPortsTTN	

```

00077 {
00078     portInconnu = 0,
00079     portMesureEnergie = 1,
00080     portMesurePoids,
00081     portMesureRuche,
00082     portMesureEnvironnement,
00083     portMesureEnsoleillement,
00084     portVol,
00085     nbPortsTTN
00086 } PortsTTN;

```

## 10.37.3.2 SeuilsAlertes

```
enum SeuilsAlertes
```

## Valeurs énumérées

tropHaut	
tropBas	
bon	

```

00056 {
00057     tropHaut = 0,
00058     tropBas = 1,
00059     bon = 2,
00060
00061 } SeuilsAlertes;

```

## 10.37.3.3 TypeAlertes

```
enum TypeAlertes
```

## Valeurs énumérées

alerteTemperatureInterieur	
alerteTemperatureExterieur	
alerteHumiditeInterieur	
alerteHumiditeExterieur	
alertePressionAtmospherique	
alertePoids	
alerteEnsoleillement	
alerteBatterie	
toutesLesAlertes	

```

00064 {
00065     alerteTemperatureInterieur = 0,
00066     alerteTemperatureExterieur = 1,
00067     alerteHumiditeInterieur = 2,
00068     alerteHumiditeExterieur = 3,
00069     alertePressionAtmospherique = 4,
00070     alertePoids = 5,
00071     alerteEnsoleillement = 6,
00072     alerteBatterie = 7,
00073     toutesLesAlertes = 8,
00074 } TypeAlertes;

```

### 10.38 Référence du fichier ParametresHoneyBeeActivity.java

#### Classes

— class [fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity](#)  
*Activité de paramétrage de l'application.*

#### Paquetages

— package [fr.campus.laurainc.honeybee](#)

### 10.39 Référence du fichier README.md

### 10.40 Référence du fichier reglagesAlertesIhm.cpp

Définition de la classe [ReglagesAlertesIhm](#).

```

#include "reglagesAlertesIhm.h"
#include "ui_reglagesAlertesIhm.h"
#include "baseDeDonnees.h"
#include <QDebug>

```

#### 10.40.1 Description détaillée

##### Auteur

Florentin Mellah, Enzo Rossi

##### Version

1.1

### 10.41 Référence du fichier reglagesAlertesIhm.h

La classe [ReglagesAlertesIhm](#).

```

#include <QDialog>
#include "alertes.h"

```

## Classes

- class [ReglagesAlertesIhm](#)  
*Déclaration de la classe [ReglagesAlertesIhm](#).*

## Espaces de nommage

- [Ui](#)

### 10.41.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.42 Référence du fichier ruche.cpp

Définition de la classe [Ruche](#).

```
#include "ruche.h"
#include "parametres.h"
#include "infosEnsoleillement.h"
#include "infosHumidite.h"
#include "infosPressionAtmospherique.h"
#include "infosTemperature.h"
#include "infosPoids.h"
#include "infosBatterie.h"
#include "alertes.h"
#include "communication.h"
#include "baseDeDonnees.h"
#include <QTextCodec>
#include <QDebug>
#include <QJsonDocument>
#include <QJsonObject>
```

### 10.42.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.43 Référence du fichier ruche.h

Déclaration de la classe [Ruche](#).

```
#include <QString>
#include <QtMqtt/QtMqtt>
#include <QtMqtt/QMqttClient>
#include "parametres.h"
```

### Classes

- struct [DonneesBatterie](#)  
*structure de données pour les mesures de la batterie*
- struct [DonneesRuche](#)  
*structure de données pour les mesures horodatées*
- struct [MesureHoraireRuche](#)  
*structure de données pour les mesures horaires*
- struct [MesureHoraireEnvironnement](#)  
*structure de données pour les mesures horaires*
- struct [MesuresHoraireEnsoleillement](#)  
*structure de données pour les mesures horaires*
- class [Ruche](#)  
*La classe [Ruche](#).*

### 10.43.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.44 Référence du fichier Ruche.java

### Classes

- class [fr.campus.laurainc.honeybee.Ruche](#)

### Paquetages

- package [fr.campus.laurainc.honeybee](#)

## 10.45 Référence du fichier ruchelhm.cpp

Définition de la classe [NouvelleRuchelhm](#).

```
#include "rucheIhm.h"
#include "ui_rucheIhm.h"
#include "ruche.h"
#include "baseDeDonnees.h"
#include "nouvelleRucheIhm.h"
#include "alertes.h"
#include "reglagesAlertesIhm.h"
#include <qwt_color_map.h>
#include <qwt_thermo.h>
#include <qwt_plot.h>
#include <qwt_plot_grid.h>
#include <qwt_plot_curve.h>
#include <qwt_symbol.h>
#include <qwt_legend.h>
#include <qwt_samples.h>
#include <qwt_plot_item.h>
#include <qwt_plot_marker.h>
#include <QDebug>
```

### 10.45.1 Description détaillée

#### Auteur

Florentin Mellah, Enzo Rossi

#### Version

1.1

## 10.46 Référence du fichier ruchelhm.h

Déclaration de la classe [Ruchelhm](#).

```
#include <QtWidgets>
#include <QVector>
#include <qwt_plot_marker.h>
#include "alertes.h"
```

#### Classes

— class [Ruchelhm](#)  
*La classe IHM.*

#### Espaces de nommage

— [Ui](#)

#### 10.46.1 Description détaillée

##### Auteur

Florentin Mellah, Enzo Rossi

##### Version

1.1

#### 10.47 Référence du fichier TimeAsXAxisLabelFormatter.java

##### Classes

— class [fr.campus.laurainc.honeybee.TimeAsXAxisLabelFormatter](#)

##### Paquetages

— package [fr.campus.laurainc.honeybee](#)

## Index

- ~Alertes
  - Alertes, [19](#)
- ~BaseDeDonnees
  - BaseDeDonnees, [44](#)
- ~Communication
  - Communication, [79](#)
- ~InfosBatterie
  - InfosBatterie, [123](#)
- ~InfosPoids
  - InfosPoids, [144](#)
- ~InfosPressionAtmospherique
  - InfosPressionAtmospherique, [148](#)
- ~InfosTemperature
  - InfosTemperature, [155](#)
- ~NouvelleRucheIhm
  - NouvelleRucheIhm, [183](#)
- ~ReglagesAlertesIhm
  - ReglagesAlertesIhm, [190](#)
- ~Ruche
  - Ruche, [234](#)
- ~RucheIhm
  - RucheIhm, [266](#)
- APP\_TITRE
  - parametres.h, [309](#)
- abonnement
  - Communication, [82](#)
- activite
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [62](#)
- addition\_isCorrect
  - fr : :campus : :laurainc : :honeybee : :ExampleUnitTest, [105](#)
- afficherAlertesBatterie
  - RucheIhm, [266](#)
- afficherAlertesEnsoleillement
  - RucheIhm, [267](#)
- afficherAlertesHumiditeExterieur
  - RucheIhm, [268](#)
- afficherAlertesHumiditeInterieur
  - RucheIhm, [269](#)
- afficherAlertesPoids
  - RucheIhm, [269](#)
- afficherAlertesPressionAtmospherique
  - RucheIhm, [270](#)
- afficherAlertesTemperatureExterieur
  - RucheIhm, [270](#)
- afficherAlertesTemperatureInterieur
  - RucheIhm, [271](#)
- afficherBatterie
  - RucheIhm, [272](#)
- afficherCharge
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [87](#)
- afficherEnsoleillement
  - RucheIhm, [272](#)
- afficherHumiditeExterieur
  - RucheIhm, [272](#)
- afficherHumiditeInterieur
  - RucheIhm, [273](#)
- afficherInformationsRuche
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [88](#)
- afficherInfosGenerale
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [88](#)
- afficherListeRuches
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [88](#)
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [107](#)
- afficherLocalisationPasserelle
  - RucheIhm, [273](#)
- afficherMesuresExt
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [89](#)
- afficherMesuresInt
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [90](#)
- afficherMesuresJournalieresEnsoleillement
  - RucheIhm, [273](#)
- afficherMesuresJournalieresEnvironnement
  - RucheIhm, [274](#)
- afficherMesuresJournalieresRuche
  - RucheIhm, [274](#)
- afficherMesuresJournalieresSelectionnee
  - RucheIhm, [275](#)
- afficherPoids
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [90](#)
  - RucheIhm, [277](#)
- afficherPressionAtmospherique
  - RucheIhm, [277](#)
- afficherTemperatureExterieur
  - RucheIhm, [277](#)
- afficherTemperatureInterieur
  - RucheIhm, [278](#)
- Alertes, [16](#)
  - ~Alertes, [19](#)
  - Alertes, [18](#)
  - alertesBatterie, [19](#)
  - alertesEnsoleillement, [20](#)
  - alertesHumiditeExterieur, [20](#)
  - alertesHumiditeInterieur, [21](#)
  - alertesPoids, [22](#)
  - alertesPressionAtmospherique, [22](#)
  - alertesTemperatureExterieur, [23](#)
  - alertesTemperatureInterieur, [24](#)
  - appelerLesAlertes, [25](#)
  - bdd, [37](#)
  - envoiAlertesBatterie, [25](#)
  - envoiAlertesEnsoleillement, [26](#)
  - envoiAlertesHumiditeExterieur, [26](#)
  - envoiAlertesHumiditeInterieur, [26](#)
  - envoiAlertesPoids, [26](#)
  - envoiAlertesPressionAtmospherique, [26](#)
  - envoiAlertesTemperatureExterieur, [26](#)
  - envoiAlertesTemperatureInterieur, [27](#)
  - envoyerMailAlerte, [27](#)
  - getIdRuche, [27](#)
  - getSeuilBatterie, [27](#)
  - getSeuilEnsoleillement, [28](#)
  - getSeuilHumiditeExterieurMax, [28](#)
  - getSeuilHumiditeExterieurMin, [28](#)
  - getSeuilHumiditeInterieurMax, [28](#)
  - getSeuilHumiditeInterieurMin, [29](#)

- getSeuilPoidsMax, [29](#)
- getSeuilPoidsMin, [29](#)
- getSeuilPressionAtmospherique, [29](#)
- getSeuilTemperatureExterieurMax, [30](#)
- getSeuilTemperatureExterieurMin, [30](#)
- getSeuilTemperatureInterieurMax, [30](#)
- getSeuilTemperatureInterieurMin, [30](#)
- idRuche, [37](#)
- infosBatterie, [37](#)
- infosEnsoleillement, [37](#)
- infosHumidite, [37](#)
- infosPoids, [37](#)
- infosPressionAtmospherique, [37](#)
- infosTemperature, [38](#)
- nomRuche, [38](#)
- recevoirSeuilBatterie, [31](#)
- recevoirSeuilEnsoleillement, [31](#)
- recevoirSeuilHumiditeExterieurMax, [31](#)
- recevoirSeuilHumiditeExterieurMin, [32](#)
- recevoirSeuilHumiditeInterieurMax, [32](#)
- recevoirSeuilHumiditeInterieurMin, [32](#)
- recevoirSeuilPoidsMax, [33](#)
- recevoirSeuilPoidsMin, [33](#)
- recevoirSeuilPressionAtmospherique, [33](#)
- recevoirSeuilTemperatureExterieurMax, [34](#)
- recevoirSeuilTemperatureExterieurMin, [34](#)
- recevoirSeuilTemperatureInterieurMax, [34](#)
- recevoirSeuilTemperatureInterieurMin, [35](#)
- setInfosBatterie, [35](#)
- setInfosEnsoleillement, [35](#)
- setInfosHumidite, [35](#)
- setInfosPoids, [36](#)
- setInfosPressionAtmospherique, [36](#)
- setInfosTemperature, [36](#)
- seuilBatterie, [38](#)
- seuilEnsoleillement, [38](#)
- seuilHumiditeExterieurMax, [38](#)
- seuilHumiditeExterieurMin, [38](#)
- seuilHumiditeInterieurMax, [38](#)
- seuilHumiditeInterieurMin, [39](#)
- seuilPoidsMax, [39](#)
- seuilPoidsMin, [39](#)
- seuilPressionAtmospherique, [39](#)
- seuilTemperatureExterieurMax, [39](#)
- seuilTemperatureExterieurMin, [39](#)
- seuilTemperatureInterieurMax, [40](#)
- seuilTemperatureInterieurMin, [40](#)
- seuilsAlertes, [39](#)
- alertes
  - ReglagesAlertesIhm, [195](#)
  - Ruche, [260](#)
- alertes.cpp, [296](#)
- alertes.h, [296](#)
- alertesActivity.java, [297](#)
- alertesBatterie
  - Alertes, [19](#)
- alertesEnsoleillement
  - Alertes, [20](#)
- alertesHumiditeExterieur
  - Alertes, [20](#)
- alertesHumiditeInterieur
  - Alertes, [21](#)
- alertesPoids
  - Alertes, [22](#)
- alertesPressionAtmospherique
  - Alertes, [22](#)
- alertesTemperatureExterieur
  - Alertes, [23](#)
- alertesTemperatureInterieur
  - Alertes, [24](#)
- annuler
  - fr : :campus : :laurainc : :honeybee : :ParametresHoney↔  
BeeActivity, [186](#)
- annulerDonneesNouvelleRuche
  - NouvelleRucheIhm, [183](#)
- AppID
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity,  
[177](#)
- appelerLesAlertes
  - Alertes, [25](#)
- arrondir
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [68](#)
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [108](#)
- BATTERIE\_SEUIL\_MIN
  - parametres.h, [309](#)
- BDD\_NOMBASE
  - parametres.h, [309](#)
- BDD\_PASSWORD
  - parametres.h, [309](#)
- BDD\_SERVEUR
  - parametres.h, [309](#)
- BDD\_USERNAME
  - parametres.h, [310](#)
- BDD
  - fr : :campus : :laurainc : :honeybee : :HoneyBee, [120](#)
- BaseDeDonnees, [43](#)
  - ~BaseDeDonnees, [44](#)
  - BaseDeDonnees, [44](#)
  - baseDeDonnees, [51](#)
  - connecter, [45](#)
  - db, [52](#)
  - detruireInstance, [45](#)
  - estConnecte, [46](#)
  - estOuvert, [46](#)
  - executer, [46](#)
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [54](#),  
[55](#)
  - getInstance, [47](#)
  - mutex, [52](#)
  - nbAcces, [52](#)
  - ouvrir, [47](#)
  - recuperer, [48–51](#)
  - typeBase, [52](#)
- baseDeDonnees
  - BaseDeDonnees, [51](#)
- baseDeDonnees.cpp, [297](#)
- baseDeDonnees.h, [297](#)
- BaseDeDonnees.java, [298](#)
- basename
  - fr : :campus : :laurainc : :honeybee : :MainActivity, [165](#)



- fr : :campus : :laurainc : :honeybee : :ParametresHoneyBeeActivity, [187](#)
- bdd
  - Alertes, [37](#)
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [62](#)
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [94](#)
  - fr : :campus : :laurainc : :honeybee : :MainActivity, [166](#)
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, [178](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [223](#)
  - NouvelleRucheIhm, [184](#)
  - ReglagesAlertesIhm, [195](#)
  - Ruche, [261](#)
  - RucheIhm, [292](#)
- boutonSupprimerRuche
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [94](#)
- btn\_seConnecter
  - fr : :campus : :laurainc : :honeybee : :MainActivity, [166](#)
- btnAjouterRuche
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, [178](#)
- btnSupprimerRuche
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, [178](#)
- calculerMoyenneHoraireEnsoleillement
  - InfosEnsoleillement, [129](#)
- calculerMoyenneHoraireHumiditeExterieur
  - InfosHumidite, [136](#)
- calculerMoyenneHoraireHumiditeInterieur
  - InfosHumidite, [136](#)
- calculerMoyenneHorairePressionAtmospherique
  - InfosPressionAtmospherique, [148](#)
- calculerMoyenneHoraireTemperatureExterieur
  - InfosTemperature, [155](#)
- calculerMoyenneHoraireTemperatureInterieur
  - InfosTemperature, [156](#)
- cardView\_Ruche
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [94](#)
- carte
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [94](#)
- CarteActivity.java, [298](#)
- Changelog.md, [298](#)
- changementEtatTTN
  - Communication, [79](#)
- charge
  - DonneesBatterie, [99](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [223](#)
  - InfosBatterie, [127](#)
- chargeEnvoye
  - InfosBatterie, [124](#)
- choixApplID
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, [178](#)
- choixRuche
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [94](#)
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [113](#)
- clearSelection
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [108](#)
- client
  - Communication, [82](#)
- clientId
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [75](#)
- ClientMQTT.java, [298](#)
- ClientMQTT
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [67](#)
- clientMQTT
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [94](#)
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [113](#)
- Communication, [77](#)
  - ~Communication, [79](#)
  - abonnement, [82](#)
  - changementEtatTTN, [79](#)
  - client, [82](#)
  - Communication, [78](#)
  - connecteTTN, [80](#)
  - connecterTTN, [79](#)
  - deconnecteTTN, [80](#)
  - deconnecterTTN, [80](#)
  - donneesRucheTTN, [82](#)
  - erreurClientTTN, [81](#)
  - etatClientConnexion, [81](#)
  - messageRecu, [81](#)
  - nomTopic, [82](#)
  - portsTTN, [83](#)
  - receptionner, [81](#)
  - reponsePingTTN, [82](#)
- communication.cpp, [299](#)
- communication.h, [299](#)
- communicationRuche
  - Ruche, [261](#)
- communiquerTTN
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [90](#)
- connecte
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [75](#)
- connecteTTN
  - Communication, [80](#)
- connecter
  - BaseDeDonnees, [45](#)
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [56](#)
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [68](#)
- connecterGestionMessageBrut
  - Ruche, [235](#)
- connecterGestionMessageTraite
  - Ruche, [235](#)
- connecterTTN
  - Communication, [79](#)
- connexion
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [63](#)
- courant
  - DonneesBatterie, [99](#)
  - InfosBatterie, [127](#)
- courantEnvoye
  - InfosBatterie, [124](#)
- courbeEnsoleillement
  - RucheIhm, [293](#)
- courbeHumiditeExterieur
  - RucheIhm, [293](#)
- courbeHumiditeInterieur
  - RucheIhm, [293](#)
- courbePressionAtmospherique
  - RucheIhm, [293](#)

- courbeTemperatureExterieur
  - RucheIhm, 293
- courbeTemperatureInterieur
  - RucheIhm, 293
- creerNouvelleRuche
  - RucheIhm, 278
- DashboardActivity.java, 299
- dateDeMiseEnService
  - fr : :campus : :laurainc : :honeybee : :Ruche, 224
- db
  - BaseDeDonnees, 52
- deconnecteTTN
  - Communication, 80
- deconnecter
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, 56
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, 68
- deconnecterSignaux
  - RucheIhm, 278
- deconnecterTTN
  - Communication, 80
- description
  - fr : :campus : :laurainc : :honeybee : :Ruche, 224
- desinscrireTopic
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, 69
- DetailsRucheActivity.java, 300
- detruireInstance
  - BaseDeDonnees, 45
- deviceId
  - fr : :campus : :laurainc : :honeybee : :Ruche, 224
- doInBackground
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees : :↔
  - ConnexionMySQL, 83
- DonneesBatterie, 98
  - charge, 99
  - courant, 99
  - horodatagecharge, 99
  - horodatagecourant, 99
  - horodatagetension, 99
  - tension, 99
- donneesBatterie
  - DonneesRuche, 101
- DonneesRuche, 100
  - donneesBatterie, 101
  - ensoleillement, 101
  - horodatageEnsoleillement, 101
  - horodatageHumiditeExterieur, 101
  - horodatageHumiditeInterieur, 102
  - horodatagePoids, 102
  - horodatagePressionAtmospherique, 102
  - horodatageTemperatureExterieur, 102
  - horodatageTemperatureInterieur, 102
  - humiditeExterieur, 102
  - humiditeInterieur, 102
  - poids, 103
  - pressionAtmospherique, 103
  - temperatureExterieur, 103
  - temperatureInterieur, 103
- donneesRuche
  - Ruche, 261
- donneesRucheTTN
  - Communication, 82
  - Ruche, 261
- donneesTTN
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, 94
- ENSOLEILLEMENT\_SEUIL\_MAX
  - parametres.h, 310
- ENSOLEILLEMENT\_SEUIL\_MIN
  - parametres.h, 310
- ed\_password
  - fr : :campus : :laurainc : :honeybee : :MainActivity, 166
- ed\_username
  - fr : :campus : :laurainc : :honeybee : :MainActivity, 166
- edAdresse
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 178
- edDescription
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 178
- edDeviceID
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 178
- edLatitude
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 178
- edLongitude
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 178
- edNomRuche
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 179
- effacerTableau
  - RucheIhm, 279
- ensoleillement
  - DonneesRuche, 101
  - fr : :campus : :laurainc : :honeybee : :Ruche, 224
  - InfosEnsoleillement, 132
- ensoleillement\_Basse
  - fr : :campus : :laurainc : :honeybee : :Ruche, 224
- ensoleillement\_Haute
  - fr : :campus : :laurainc : :honeybee : :Ruche, 224
- ensoleillement\_Moyenne
  - fr : :campus : :laurainc : :honeybee : :Ruche, 224
- ensoleillement\_afficher
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 113
- ensoleillementEnvoye
  - InfosEnsoleillement, 130
- ensoleillementRuche
  - Ruche, 261
- ensoleillements
  - InfosEnsoleillement, 132
- ensoleillementsMaximum
  - MesuresHoraireEnsoleillement, 173
- ensoleillementsMinimum
  - MesuresHoraireEnsoleillement, 173
- ensoleillementsMoyen
  - MesuresHoraireEnsoleillement, 173
- ensoleillementsRecues
  - MesuresHoraireEnsoleillement, 173
- envoiAlertesBatterie
  - Alertes, 25

- Ruche, [236](#)
- envoiAlertesEnsoleillement
  - Alertes, [26](#)
  - Ruche, [236](#)
- envoiAlertesHumiditeExterieur
  - Alertes, [26](#)
  - Ruche, [236](#)
- envoiAlertesHumiditeInterieur
  - Alertes, [26](#)
  - Ruche, [236](#)
- envoiAlertesPoids
  - Alertes, [26](#)
  - Ruche, [236](#)
- envoiAlertesPressionAtmospherique
  - Alertes, [26](#)
  - Ruche, [237](#)
- envoiAlertesTemperatureExterieur
  - Alertes, [26](#)
  - Ruche, [237](#)
- envoiAlertesTemperatureInterieur
  - Alertes, [27](#)
  - Ruche, [237](#)
- envoiLocalisationPasserelle
  - Ruche, [237](#)
- envoiSeuilEnsoleillementMin
  - ReglagesAlertesIhm, [191](#)
- envoiSeuilHumiditeExterieurMax
  - ReglagesAlertesIhm, [191](#)
- envoiSeuilHumiditeExterieurMin
  - ReglagesAlertesIhm, [191](#)
- envoiSeuilHumiditeInterieurMax
  - ReglagesAlertesIhm, [191](#)
- envoiSeuilHumiditeInterieurMin
  - ReglagesAlertesIhm, [191](#)
- envoiSeuilPoidsMax
  - ReglagesAlertesIhm, [192](#)
- envoiSeuilPoidsMin
  - ReglagesAlertesIhm, [192](#)
- envoiSeuilPressionAtmospheriqueMin
  - ReglagesAlertesIhm, [192](#)
- envoiSeuilTemperatureExterieurMax
  - ReglagesAlertesIhm, [192](#)
- envoiSeuilTemperatureExterieurMin
  - ReglagesAlertesIhm, [192](#)
- envoiSeuilTemperatureInterieurMax
  - ReglagesAlertesIhm, [192](#)
- envoiSeuilTemperatureInterieurMin
  - ReglagesAlertesIhm, [193](#)
- envoyerMailAlerte
  - Alertes, [27](#)
- erreurClientTTN
  - Communication, [81](#)
- estConnecte
  - BaseDeDonnees, [46](#)
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [57](#)
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [69](#)
- estInscrit
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [69](#)
- estOuvert
  - BaseDeDonnees, [46](#)
- etatClientConnexion
  - Communication, [81](#)
  - Ruche, [237](#)
- ExampleInstrumentedTest.java, [300](#)
- ExampleUnitTest.java, [300](#)
- executer
  - BaseDeDonnees, [46](#)
- executerRequete
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [57](#)
- extraireCharge
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [70](#)
- extraireDonnees
  - Ruche, [237](#)
- extraireDonneesBrutes
  - Ruche, [239](#)
- extraireEnsoleillement
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [70](#)
- extraireHorodatage
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [70](#)
- extraireHumiditeExterieur
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [71](#)
- extraireHumiditeInterieure
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [71](#)
- extrairePoids
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [72](#)
  - Ruche, [239](#)
- extrairePression
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [72](#)
- extraireTemperatureExterieur
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [73](#)
- extraireTemperatureInterieure
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [73](#)
- fenetreSelectionDate
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, [179](#)
- formatLabel
  - fr : :campus : :laurainc : :honeybee : :TimeAsXAxisLabel←  
Formatter, [295](#)
- fr, [15](#)
- fr.campus, [15](#)
- fr.campus.laurainc, [16](#)
- fr.campus.laurainc.honeybee, [16](#)
- fr.campus.laurainc.honeybee.alertesActivity, [41](#)
- fr.campus.laurainc.honeybee.BaseDeDonnees, [53](#)
- fr.campus.laurainc.honeybee.BaseDeDonnees.ConnexionMySQL, [83](#)
- fr.campus.laurainc.honeybee.CarteActivity, [64](#)
- fr.campus.laurainc.honeybee.ClientMQTT, [66](#)
- fr.campus.laurainc.honeybee.DashboardActivity, [86](#)
- fr.campus.laurainc.honeybee.DetailsRucheActivity, [97](#)
- fr.campus.laurainc.honeybee.ExampleInstrumentedTest, [103](#)
- fr.campus.laurainc.honeybee.ExampleUnitTest, [104](#)
- fr.campus.laurainc.honeybee.GraphActivity, [106](#)
- fr.campus.laurainc.honeybee.homeActivity, [117](#)
- fr.campus.laurainc.honeybee.HoneyBee, [120](#)
- fr.campus.laurainc.honeybee.MainActivity, [162](#)
- fr.campus.laurainc.honeybee.NouvelleRucheActivity, [174](#)
- fr.campus.laurainc.honeybee.ParametresHoneyBeeActivity, [185](#)
- fr.campus.laurainc.honeybee.Ruche, [196](#)
- fr.campus.laurainc.honeybee.TimeAsXAxisLabelFormatter, [295](#)
- fr : :campus : :laurainc : :honeybee : :BaseDeDonnees

- activite, 62
- BaseDeDonnees, 54, 55
- bdd, 62
- connecter, 56
- connexion, 63
- deconnecter, 56
- estConnecte, 57
- executerRequete, 57
- getInstance, 58–60
- mutex, 63
- password, 63
- readResolve, 60
- reconnecter, 60
- setActivite, 61
- setParametres, 61
- supprimerRuche, 62
- TAG, 63
- url, 63
- username, 63
- fr : :campus : :laurainc : :honeybee : :BaseDeDonnees : :↵
  - ConnexionMySQL
  - doInBackground, 83
  - messageConnexion, 85
  - onPostExecute, 84
  - onProgressUpdate, 84
- fr : :campus : :laurainc : :honeybee : :CarteActivity
  - mMap, 65
  - onCreate, 64
  - onMapReady, 65
- fr : :campus : :laurainc : :honeybee : :ClientMQTT
  - arrondir, 68
  - clientId, 75
  - ClientMQTT, 67
  - connecte, 75
  - connecter, 68
  - deconnecter, 68
  - desinscrireTopic, 69
  - estConnecte, 69
  - estInscrit, 69
  - extraireCharge, 70
  - extraireEnsoleillement, 70
  - extraireHorodatage, 70
  - extraireHumiditeExterieur, 71
  - extraireHumiditeInterieur, 71
  - extrairePoids, 72
  - extrairePression, 72
  - extraireTemperatureExterieur, 73
  - extraireTemperatureInterieur, 73
  - inscrit, 75
  - mqttAndroidClient, 75
  - PORT\_MESURE\_ENERGIE, 76
  - PORT\_MESURE\_ENSOLEILLEMENT, 76
  - PORT\_MESURE\_ENVIRONNEMENT, 76
  - PORT\_MESURE\_POIDS, 76
  - PORT\_MESURE\_RUCHE, 76
  - password, 75
  - serverUri, 76
  - setCallback, 74
  - setSubscriptionTopic, 74
  - souscrireTopic, 74
  - subscriptionTopic, 76
  - TAG, 77
  - username, 77
- fr : :campus : :laurainc : :honeybee : :DashboardActivity
  - afficherCharge, 87
  - afficherInformationsRuche, 88
  - afficherInfosGenerale, 88
  - afficherListeRuches, 88
  - afficherMesuresExt, 89
  - afficherMesuresInt, 90
  - afficherPoids, 90
  - bdd, 94
  - boutonSupprimerRuche, 94
  - cardView\_Ruche, 94
  - carte, 94
  - choixRuche, 94
  - clientMQTT, 94
  - communiquerTTN, 90
  - donneesTTN, 94
  - graphique, 94
  - handler, 95
  - imgCharge, 95
  - mesRuches, 95
  - onCreate, 91
  - recupererElementsVue, 91
  - supprimerRuche, 92
  - TAG, 95
  - traiterMessage, 92
  - txtAlertes, 95
  - txtCharge, 96
  - txtDescription, 96
  - txtEnsoleillement, 96
  - txtGPS, 96
  - txtHorodatage, 96
  - txtHumiditeExt, 96
  - txtHumiditeInt, 96
  - txtPoids, 96
  - txtPression, 96
  - txtTemperatureExt, 97
  - txtTemperatureInt, 97
- fr : :campus : :laurainc : :honeybee : :DetailsRucheActivity
  - onCreate, 97
  - setIdRuche, 98
- fr : :campus : :laurainc : :honeybee : :ExampleInstrumentedTest
  - useAppContext, 104
- fr : :campus : :laurainc : :honeybee : :ExampleUnitTest
  - addition\_isCorrect, 105
- fr : :campus : :laurainc : :honeybee : :GraphActivity
  - afficherListeRuches, 107
  - arrondir, 108
  - choixRuche, 113
  - clearSelection, 108
  - clientMQTT, 113
  - ensoleillement\_afficher, 113
  - graphiqueTemperatures, 113
  - handler, 113
  - humidite\_ext\_afficher, 114
  - humidite\_int\_afficher, 114
  - initialisationElementsIHM, 108
  - initialiserCallbacks, 109
  - iv\_ensoleillement, 114
  - iv\_humidite\_ext, 114

- iv\_humidite\_int, 114
- iv\_poids, 114
- iv\_pression, 115
- iv\_tem\_ext, 115
- iv\_tem\_int, 115
- mSeriesEnsoleillement, 115
- mSeriesHumiditeExt, 115
- mSeriesHumiditeInt, 115
- mSeriesPoids, 115
- mSeriesPression, 115
- mSeriesTemperaturesExt, 116
- mSeriesTemperaturesInt, 116
- mesRuches, 115
- onCreate, 112
- poids\_afficher, 116
- pression\_afficher, 116
- ruche, 116
- TAG, 116
- temp\_ext\_afficher, 116
- temp\_int\_afficher, 116
- temps\_reel\_afficher, 116
- tv\_donneesBas, 117
- tv\_donneesHaut, 117
- tv\_donneesMoyenne, 117
- tv\_valeurCourante, 117
- fr : :campus : :laurainc : :honeybee : :HoneyBee
  - BDD, 120
  - ID\_Intent\_ParametresHoneyBee, 121
  - REQUETE\_SQL\_ALERTES, 121
  - REQUETE\_SQL\_ERREUR, 121
  - REQUETE\_SQL\_IDTTN, 121
  - REQUETE\_SQL\_LISTE\_RUCHES, 121
  - REQUETE\_SQL\_MESURES\_RUCHES, 121
  - REQUETE\_SQL\_OK, 122
  - REQUETE\_SQL\_RUCHE, 122
- fr : :campus : :laurainc : :honeybee : :MainActivity
  - basename, 165
  - bdd, 166
  - btn\_seConnecter, 166
  - ed\_password, 166
  - ed\_username, 166
  - hostname, 166
  - m\_linearLayout, 166
  - onActivityResult, 164
  - onCreate, 164
  - password, 166
  - TAG, 166
  - tv\_erreur, 166
  - url, 167
  - username, 167
- fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity
  - AppID, 177
  - bdd, 178
  - btnAjouterRuche, 178
  - btnSupprimerRuche, 178
  - choixAppID, 178
  - edAdresse, 178
  - edDescription, 178
  - edDeviceID, 178
  - edLatitude, 178
  - edLongitude, 178
  - edNomRuche, 179
  - fenetreSelectionDate, 179
  - getIdTTN, 175
  - handler, 179
  - idTTN, 179
  - initialiserChoixAppID, 175
  - initialiserMenuOnglets, 176
  - menuOnglets, 179
  - onCreate, 176
  - recupererElementsIHM, 177
  - rucheUtilitaire, 180
  - TAG, 180
  - tv\_date, 180
- fr : :campus : :laurainc : :honeybee : :ParametresHoneyBeeActivity
  - annuler, 186
  - basename, 187
  - hostname, 187
  - onCreate, 186
  - password, 187
  - TAG, 187
  - username, 187
  - valider, 186
- fr : :campus : :laurainc : :honeybee : :Ruche
  - bdd, 223
  - charge, 223
  - dateDeMiseEnService, 224
  - description, 224
  - deviceID, 224
  - ensoleillement, 224
  - ensoleillement\_Basse, 224
  - ensoleillement\_Haute, 224
  - ensoleillement\_Moyenne, 224
  - getCharge, 200
  - getDateDeMiseEnService, 200
  - getDescription, 200
  - getDeviceID, 201
  - getEnsoleillement, 201
  - getEnsoleillement\_Basse, 201
  - getEnsoleillement\_Haute, 201
  - getEnsoleillement\_Moyenne, 202
  - getHistoriqueAlertes, 202
  - getHum\_ext\_Basse, 202
  - getHum\_ext\_Haute, 202
  - getHum\_ext\_Moyenne, 203
  - getHum\_int\_Basse, 203
  - getHum\_int\_Haute, 203
  - getHum\_int\_Moyenne, 203
  - getHumiditeExt, 204
  - getHumiditeInt, 204
  - getIdRuche, 204
  - getIdTTNSelectionne, 204
  - getLatitude, 205
  - getListeChoixAppID, 205
  - getListeRuches, 205
  - getLongitude, 205
  - getNom, 207
  - getPoids, 208
  - getPoids\_Basse, 208
  - getPoids\_Haute, 208
  - getPoids\_Moyenne, 208
  - getPression, 209

- getPression\_Basse, 209
- getPression\_Haute, 209
- getPression\_Moyenne, 209
- getTemp\_ext\_Basse, 210
- getTemp\_ext\_Haute, 210
- getTemp\_ext\_Moyenne, 210
- getTemp\_int\_Basse, 210
- getTemp\_int\_Haute, 211
- getTemp\_int\_Moyenne, 211
- getTemperatureExt, 211
- getTemperatureInt, 211
- getmSeriesEnsoleillement, 206
- getmSeriesHumiditeExt, 206
- getmSeriesHumiditeInt, 206
- getmSeriesPoids, 206
- getmSeriesPression, 207
- getmSeriesTemperaturesExt, 207
- getmSeriesTemperaturesInt, 207
- handler, 225
- historiqueAlertes, 225
- Horodatage, 225
- hum\_ext\_Basse, 225
- hum\_ext\_Haute, 225
- hum\_ext\_Moyenne, 225
- hum\_int\_Basse, 225
- hum\_int\_Haute, 226
- hum\_int\_Moyenne, 226
- humiditeExt, 226
- humiditeInt, 226
- idRuche, 226
- idTTNSelectionne, 226
- latitude, 226
- listeChoixAppID, 227
- listeChoixRuches, 227
- longitude, 227
- mSeriesEnsoleillement, 227
- mSeriesHumiditeExt, 227
- mSeriesHumiditeInt, 227
- mSeriesPoids, 227
- mSeriesPression, 228
- mSeriesTemperaturesExt, 228
- mSeriesTemperaturesInt, 228
- nom, 228
- poids, 228
- poids\_Basse, 228
- poids\_Haute, 228
- poids\_Moyenne, 229
- pression, 229
- pression\_Basse, 229
- pression\_Haute, 229
- pression\_Moyenne, 229
- recuperer, 212
- recupererChoixChAppID, 214
- recupererHistoriqueAlertes, 214
- recupererIdTTN, 215
- recupererListeRuches, 216
- recupererMesuresJournalieresRuche, 217
- recupererMoyennes, 219
- Ruche, 198, 199
- setCharge, 221
- setEnsoleillement, 221
- setHandler, 221
- setHorodatage, 222
- setHumiditeExt, 222
- setHumiditeInt, 222
- setPoids, 222
- setPression, 222
- setTemperatureExt, 223
- setTemperatureInt, 223
- TAG, 229
- temp\_ext\_Basse, 229
- temp\_ext\_Haute, 230
- temp\_ext\_Moyenne, 230
- temp\_int\_Basse, 230
- temp\_int\_Haute, 230
- temp\_int\_Moyenne, 230
- temperatureExt, 230
- temperatureInt, 230
- fr : :campus : :laurainc : :honeybee : :TimeAsXAxisLabelFormatter
  - formatLabel, 295
  - mFormat, 296
  - TimeAsXAxisLabelFormatter, 295
- fr : :campus : :laurainc : :honeybee : :alertesActivity
  - handler, 42
  - onCreate, 42
  - ruche, 42
  - TAG, 42
  - tv\_hitoriqueAlertes, 43
- fr : :campus : :laurainc : :honeybee : :homeActivity
  - img\_disconnect, 119
  - layout\_alertes, 119
  - layout\_debug, 119
  - layout\_gestionAlertes, 119
  - layout\_gestionRuches, 119
  - layout\_graphs, 119
  - layout\_map, 119
  - layout\_tableauDeBord, 119
  - onCreate, 118
- gererAlertes
  - Ruche, 240
- getAlertes
  - Ruche, 240
- getCharge
  - fr : :campus : :laurainc : :honeybee : :Ruche, 200
- getDateDeMiseEnService
  - fr : :campus : :laurainc : :honeybee : :Ruche, 200
- getDescription
  - fr : :campus : :laurainc : :honeybee : :Ruche, 200
- getDeviceID
  - fr : :campus : :laurainc : :honeybee : :Ruche, 201
- getEnsoleillement
  - fr : :campus : :laurainc : :honeybee : :Ruche, 201
  - InfosEnsoleillement, 130
- getEnsoleillement\_Basse
  - fr : :campus : :laurainc : :honeybee : :Ruche, 201
- getEnsoleillement\_Haute
  - fr : :campus : :laurainc : :honeybee : :Ruche, 201
- getEnsoleillement\_Moyenne
  - fr : :campus : :laurainc : :honeybee : :Ruche, 202
- getHistoriqueAlertes
  - fr : :campus : :laurainc : :honeybee : :Ruche, 202



getHorodatageBatterie  
     InfosBatterie, [124](#)  
 getHorodatageEnsoleillement  
     InfosEnsoleillement, [130](#)  
 getHorodatageHumiditeExterieur  
     InfosHumidite, [137](#)  
 getHorodatageHumiditeInterieur  
     InfosHumidite, [137](#)  
 getHorodatagePoids  
     InfosPoids, [144](#)  
 getHorodatagePressionAtmospherique  
     InfosPressionAtmospherique, [149](#)  
 getHorodatageTemperatureExterieur  
     InfosTemperature, [156](#)  
 getHorodatageTemperatureInterieur  
     InfosTemperature, [157](#)  
 getHum\_ext\_Basse  
     fr : :campus : :laurainc : :honeybee : :Ruche, [202](#)  
 getHum\_ext\_Haute  
     fr : :campus : :laurainc : :honeybee : :Ruche, [202](#)  
 getHum\_ext\_Moyenne  
     fr : :campus : :laurainc : :honeybee : :Ruche, [203](#)  
 getHum\_int\_Basse  
     fr : :campus : :laurainc : :honeybee : :Ruche, [203](#)  
 getHum\_int\_Haute  
     fr : :campus : :laurainc : :honeybee : :Ruche, [203](#)  
 getHum\_int\_Moyenne  
     fr : :campus : :laurainc : :honeybee : :Ruche, [203](#)  
 getHumiditeExt  
     fr : :campus : :laurainc : :honeybee : :Ruche, [204](#)  
 getHumiditeExterieur  
     InfosHumidite, [137](#)  
 getHumiditeInt  
     fr : :campus : :laurainc : :honeybee : :Ruche, [204](#)  
 getHumiditeInterieur  
     InfosHumidite, [138](#)  
 getIdRuche  
     Alertes, [27](#)  
     fr : :campus : :laurainc : :honeybee : :Ruche, [204](#)  
     Ruche, [241](#)  
 getIdTTNSelectionne  
     fr : :campus : :laurainc : :honeybee : :Ruche, [204](#)  
 getIdTTN  
     fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity,  
         [175](#)  
 getInstance  
     BaseDeDonnees, [47](#)  
     fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [58–](#)  
         [60](#)  
 getLatitude  
     fr : :campus : :laurainc : :honeybee : :Ruche, [205](#)  
 getListeChoixAppID  
     fr : :campus : :laurainc : :honeybee : :Ruche, [205](#)  
 getListeRuches  
     fr : :campus : :laurainc : :honeybee : :Ruche, [205](#)  
 getLongitude  
     fr : :campus : :laurainc : :honeybee : :Ruche, [205](#)  
 getMesureBatterieCharge  
     InfosBatterie, [124](#)  
 getMesureBatterieCourant  
     InfosBatterie, [124](#)  
 getMesureBatterieTension  
     InfosBatterie, [125](#)  
 getNom  
     fr : :campus : :laurainc : :honeybee : :Ruche, [207](#)  
 getPoids  
     fr : :campus : :laurainc : :honeybee : :Ruche, [208](#)  
     InfosPoids, [145](#)  
 getPoids\_Basse  
     fr : :campus : :laurainc : :honeybee : :Ruche, [208](#)  
 getPoids\_Haute  
     fr : :campus : :laurainc : :honeybee : :Ruche, [208](#)  
 getPoids\_Moyenne  
     fr : :campus : :laurainc : :honeybee : :Ruche, [208](#)  
 getPression  
     fr : :campus : :laurainc : :honeybee : :Ruche, [209](#)  
 getPression\_Basse  
     fr : :campus : :laurainc : :honeybee : :Ruche, [209](#)  
 getPression\_Haute  
     fr : :campus : :laurainc : :honeybee : :Ruche, [209](#)  
 getPression\_Moyenne  
     fr : :campus : :laurainc : :honeybee : :Ruche, [209](#)  
 getPressionAtmospherique  
     InfosPressionAtmospherique, [149](#)  
 getSeuilBatterie  
     Alertes, [27](#)  
 getSeuilEnsoleillement  
     Alertes, [28](#)  
 getSeuilHumiditeExterieurMax  
     Alertes, [28](#)  
 getSeuilHumiditeExterieurMin  
     Alertes, [28](#)  
 getSeuilHumiditeInterieurMax  
     Alertes, [28](#)  
 getSeuilHumiditeInterieurMin  
     Alertes, [29](#)  
 getSeuilPoidsMax  
     Alertes, [29](#)  
 getSeuilPoidsMin  
     Alertes, [29](#)  
 getSeuilPressionAtmospherique  
     Alertes, [29](#)  
 getSeuilTemperatureExterieurMax  
     Alertes, [30](#)  
 getSeuilTemperatureExterieurMin  
     Alertes, [30](#)  
 getSeuilTemperatureInterieurMax  
     Alertes, [30](#)  
 getSeuilTemperatureInterieurMin  
     Alertes, [30](#)  
 getTemp\_ext\_Basse  
     fr : :campus : :laurainc : :honeybee : :Ruche, [210](#)  
 getTemp\_ext\_Haute  
     fr : :campus : :laurainc : :honeybee : :Ruche, [210](#)  
 getTemp\_ext\_Moyenne  
     fr : :campus : :laurainc : :honeybee : :Ruche, [210](#)  
 getTemp\_int\_Basse  
     fr : :campus : :laurainc : :honeybee : :Ruche, [210](#)  
 getTemp\_int\_Haute  
     fr : :campus : :laurainc : :honeybee : :Ruche, [211](#)  
 getTemp\_int\_Moyenne  
     fr : :campus : :laurainc : :honeybee : :Ruche, [211](#)

- getTemperatureExt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 211
- getTemperatureExterieur
  - InfosTemperature, 157
- getTemperatureInt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 211
- getTemperatureInterieur
  - InfosTemperature, 157
- getmSeriesEnsoleillement
  - fr : :campus : :laurainc : :honeybee : :Ruche, 206
- getmSeriesHumiditeExt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 206
- getmSeriesHumiditeInt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 206
- getmSeriesPoids
  - fr : :campus : :laurainc : :honeybee : :Ruche, 206
- getmSeriesPression
  - fr : :campus : :laurainc : :honeybee : :Ruche, 207
- getmSeriesTemperaturesExt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 207
- getmSeriesTemperaturesInt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 207
- GraphActivity.java, 300
- graphique
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, 94
- graphiqueTemperatures
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 113
- HUMIDITE\_EXTERIEUR\_SEUIL\_MAX
  - parametres.h, 310
- HUMIDITE\_EXTERIEUR\_SEUIL\_MIN
  - parametres.h, 310
- HUMIDITE\_INTERIEUR\_SEUIL\_MAX
  - parametres.h, 310
- HUMIDITE\_INTERIEUR\_SEUIL\_MIN
  - parametres.h, 310
- handler
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, 95
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 113
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 179
  - fr : :campus : :laurainc : :honeybee : :Ruche, 225
  - fr : :campus : :laurainc : :honeybee : :alertesActivity, 42
- heure
  - MesureHoraireEnvironement, 168
  - MesureHoraireRuche, 171
  - MesuresHoraireEnsoleillement, 173
- heureCouranteEnsoleillement
  - InfosEnsoleillement, 133
- heureCouranteHumiditeExterieur
  - InfosHumidite, 141
- heureCouranteHumiditeInterieur
  - InfosHumidite, 142
- heureCourantePressionAtmospherique
  - InfosPressionAtmospherique, 151
- heureCouranteTemperatureExterieur
  - InfosTemperature, 161
- heureCouranteTemperatureInterieur
  - InfosTemperature, 161
- historiqueAlertes
  - fr : :campus : :laurainc : :honeybee : :Ruche, 225
- homeActivity.java, 300
- HoneyBee.java, 301
- Horodatage
  - fr : :campus : :laurainc : :honeybee : :Ruche, 225
- horodatageBatterie
  - InfosBatterie, 127
- horodatageEnsoleillement
  - DonneesRuche, 101
  - InfosEnsoleillement, 133
- horodatageHumiditeExterieur
  - DonneesRuche, 101
  - InfosHumidite, 142
- horodatageHumiditeInterieur
  - DonneesRuche, 102
  - InfosHumidite, 142
- horodatagePoids
  - DonneesRuche, 102
  - InfosPoids, 146
- horodatagePressionAtmospherique
  - DonneesRuche, 102
  - InfosPressionAtmospherique, 151
- horodatageTemperatureExterieur
  - DonneesRuche, 102
  - InfosTemperature, 161
- horodatageTemperatureInterieur
  - DonneesRuche, 102
  - InfosTemperature, 161
- horodatagecharge
  - DonneesBatterie, 99
- horodatagecourant
  - DonneesBatterie, 99
- horodatagetension
  - DonneesBatterie, 99
- hostname
  - fr : :campus : :laurainc : :honeybee : :MainActivity, 166
  - fr : :campus : :laurainc : :honeybee : :ParametresHoneyBeeActivity, 187
- hum\_ext\_Basse
  - fr : :campus : :laurainc : :honeybee : :Ruche, 225
- hum\_ext\_Haute
  - fr : :campus : :laurainc : :honeybee : :Ruche, 225
- hum\_ext\_Moyenne
  - fr : :campus : :laurainc : :honeybee : :Ruche, 225
- hum\_int\_Basse
  - fr : :campus : :laurainc : :honeybee : :Ruche, 225
- hum\_int\_Haute
  - fr : :campus : :laurainc : :honeybee : :Ruche, 226
- hum\_int\_Moyenne
  - fr : :campus : :laurainc : :honeybee : :Ruche, 226
- humidite\_ext\_afficher
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 114
- humidite\_int\_afficher
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 114
- humiditeExt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 226
- humiditeExterieur
  - DonneesRuche, 102
  - InfosHumidite, 142
- humiditeExterieurEnvoye
  - InfosHumidite, 138
- humiditeExterieurMaximum



- MesureHoraireEnvironement, 168
- humiditeExterieurMinimum
  - MesureHoraireEnvironement, 168
- humiditeExterieurMoyenne
  - MesureHoraireEnvironement, 168
- humiditeExterieurs
  - InfosHumidite, 142
- humiditeInt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 226
- humiditeInterieur
  - DonneesRuche, 102
  - InfosHumidite, 142
- humiditeInterieurEnvoye
  - InfosHumidite, 138
- humiditeInterieurMaximum
  - MesureHoraireRuche, 171
- humiditeInterieurMinimum
  - MesureHoraireRuche, 171
- humiditeInterieurMoyenne
  - MesureHoraireRuche, 171
- humiditeInterieurs
  - InfosHumidite, 142
- humiditeRuche
  - Ruche, 261
- humiditesRecues
  - MesureHoraireEnvironement, 169
  - MesureHoraireRuche, 171
- ID\_Intent\_ParametresHoneyBee
  - fr : :campus : :laurainc : :honeybee : :HoneyBee, 121
- INSTALL.md, 306
- idRuche
  - Alertes, 37
  - fr : :campus : :laurainc : :honeybee : :Ruche, 226
- idTTNSelectionne
  - fr : :campus : :laurainc : :honeybee : :Ruche, 226
- idTTN
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 179
- img\_disconnect
  - fr : :campus : :laurainc : :honeybee : :homeActivity, 119
- imgCharge
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, 95
- InfosBatterie, 122
  - ~InfosBatterie, 123
  - charge, 127
  - chargeEnvoye, 124
  - courant, 127
  - courantEnvoye, 124
  - getHorodatageBatterie, 124
  - getMesureBatterieCharge, 124
  - getMesureBatterieCourant, 124
  - getMesureBatterieTension, 125
  - horodatageBatterie, 127
  - InfosBatterie, 123
  - setHorodatageBatterie, 125
  - setMesuresBatterie, 125
  - tension, 127
  - tensionEnvoye, 125
  - traiterNouvelleMesureCharge, 126
  - traiterNouvelleMesureCourant, 126
  - traiterNouvelleMesureTension, 126
- infosBatterie
  - Alertes, 37
  - Ruche, 262
- infosBatterie.cpp, 301
- infosBatterie.h, 301
- InfosEnsoleillement, 128
  - calculerMoyenneHoraireEnsoleillement, 129
  - ensoleillement, 132
  - ensoleillementEnvoye, 130
  - ensoleillements, 132
  - getEnsoleillement, 130
  - getHorodatageEnsoleillement, 130
  - heureCouranteEnsoleillement, 133
  - horodatageEnsoleillement, 133
  - InfosEnsoleillement, 129
  - setEnsoleillement, 131
  - setHorodatageEnsoleillement, 131
  - traitementEnsoleillementEnvoye, 131
  - traiterNouvelleEnsoleillement, 132
- infosEnsoleillement
  - Alertes, 37
- infosEnsoleillement.cpp, 302
- infosEnsoleillement.h, 302
- InfosHumidite, 133
  - calculerMoyenneHoraireHumiditeExterieur, 136
  - calculerMoyenneHoraireHumiditeInterieur, 136
  - getHorodatageHumiditeExterieur, 137
  - getHorodatageHumiditeInterieur, 137
  - getHumiditeExterieur, 137
  - getHumiditeInterieur, 138
  - heureCouranteHumiditeExterieur, 141
  - heureCouranteHumiditeInterieur, 142
  - horodatageHumiditeExterieur, 142
  - horodatageHumiditeInterieur, 142
  - humiditeExterieur, 142
  - humiditeExterieurEnvoye, 138
  - humiditeExterieurs, 142
  - humiditeInterieur, 142
  - humiditeInterieurEnvoye, 138
  - humiditeInterieurs, 142
  - InfosHumidite, 135
  - setHorodatageHumiditeExterieur, 138
  - setHorodatageHumiditeInterieur, 139
  - setHumiditeExterieur, 139
  - setHumiditeInterieur, 139
  - traitementHumiditeExterieurEnvoye, 140
  - traitementHumiditeInterieurEnvoye, 140
  - traiterNouvelleHumiditeExterieur, 140
  - traiterNouvelleHumiditeInterieur, 141
- infosHumidite
  - Alertes, 37
- infosHumidite.cpp, 303
- infosHumidite.h, 303
- InfosPoids, 143
  - ~InfosPoids, 144
  - getHorodatagePoids, 144
  - getPoids, 145
  - horodatagePoids, 146
  - InfosPoids, 144
  - poids, 146

- poidsEnvoye, 145
- setHorodatagePoids, 145
- setPoids, 145
- traiterNouveauPoids, 146
- infosPoids
  - Alertes, 37
  - Ruche, 262
- infosPoids.cpp, 303
- infosPoids.h, 304
- InfosPressionAtmospherique, 147
  - ~InfosPressionAtmospherique, 148
  - calculerMoyenneHorairePressionAtmospherique, 148
  - getHorodatagePressionAtmospherique, 149
  - getPressionAtmospherique, 149
  - heureCourantePressionAtmospherique, 151
  - horodatagePressionAtmospherique, 151
  - InfosPressionAtmospherique, 148
  - pressionAtmospherique, 151
  - pressionAtmospheriqueEnvoye, 149
  - pressionAtmospheriques, 152
  - setHorodatagePressionAtmospherique, 150
  - setPressionAtmospherique, 150
  - traitementPressionAtmospheriqueEnvoye, 150
  - traiterNouvellePressionAtmospherique, 150
- infosPressionAtmospherique
  - Alertes, 37
- infosPressionAtmospherique.cpp, 304
- infosPressionAtmospherique.h, 305
- InfosTemperature, 152
  - ~InfosTemperature, 155
  - calculerMoyenneHoraireTemperatureExterieur, 155
  - calculerMoyenneHoraireTemperatureInterieur, 156
  - getHorodatageTemperatureExterieur, 156
  - getHorodatageTemperatureInterieur, 157
  - getTemperatureExterieur, 157
  - getTemperatureInterieur, 157
  - heureCouranteTemperatureExterieur, 161
  - heureCouranteTemperatureInterieur, 161
  - horodatageTemperatureExterieur, 161
  - horodatageTemperatureInterieur, 161
  - InfosTemperature, 155
  - setHorodatageTemperatureExterieur, 157
  - setHorodatageTemperatureInterieur, 158
  - setTemperatureExterieur, 158
  - setTemperatureInterieur, 158
  - temperatureExterieur, 161
  - temperatureExterieurEnvoye, 158
  - temperatureInterieur, 162
  - temperatureInterieurEnvoye, 159
  - temperaturesExterieurs, 162
  - temperaturesInterieurs, 162
  - traitementTemperatureExterieurEnvoye, 159
  - traitementTemperatureInterieurEnvoye, 159
  - traiterMoyenneHoraireTemperatureInterieur, 159
  - traiterNouvelleTemperatureExterieur, 160
  - traiterNouvelleTemperatureInterieur, 160
- infosTemperature
  - Alertes, 38
- infosTemperature.cpp, 305
- infosTemperature.h, 305
- initialisationElementsIHM
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 108
- initialiserCallbacks
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 109
- initialiserChoixAppID
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 175
- initialiserEnsoleillementCourbe
  - Ruchelhm, 280
- initialiserHumiditeExterieurCourbe
  - Ruchelhm, 280
- initialiserHumiditeInterieurCourbe
  - Ruchelhm, 280
- initialiserMarqueurAlerteEnsoleillement
  - Ruchelhm, 281
- initialiserMarqueurAlerteHumiditeExterieur
  - Ruchelhm, 281
- initialiserMarqueurAlerteHumiditeInterieur
  - Ruchelhm, 281
- initialiserMarqueurAlertePression
  - Ruchelhm, 282
- initialiserMarqueurAlerteTemperatureExterieur
  - Ruchelhm, 282
- InitialiserMarqueurAlerteTemperatureInterieur
  - Ruchelhm, 282
- initialiserMenuOnglets
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 176
- initialiserPressionAtmospheriqueCourbe
  - Ruchelhm, 283
- initialiserTemperatureExterieurCourbe
  - Ruchelhm, 283
- initialiserTemperatureInterieurCourbe
  - Ruchelhm, 283
- initialiserThermometre
  - Ruchelhm, 284
- initialiserWidgets
  - Ruchelhm, 284
- initialiserCourbes
  - Ruchelhm, 285
- inscrit
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, 75
- insérerDonneesPortBatterie
  - Ruche, 241
- insérerDonneesPortEnsoleillement
  - Ruche, 241
- insérerDonneesPortMesureEnvironnement
  - Ruche, 242
- insérerDonneesPortMesureRuche
  - Ruche, 242
- insérerDonneesPortPoids
  - Ruche, 242
- insérerMesureHoraireBatterie
  - Ruche, 243
- insérerMesureHoraireEnsoleillement
  - Ruche, 243
- insérerMesureHoraireEnvironnement
  - Ruche, 243
- insérerMesureHorairePoids
  - Ruche, 244
- insérerMesureHoraireRuche
  - Ruche, 244

- insererMesuresEnsoleillementTableau
  - Ruchelhm, [286](#)
- insererMesuresEnvironnementTableau
  - Ruchelhm, [287](#)
- insererMesuresRucheTableau
  - Ruchelhm, [287](#)
- iv\_ensembleillement
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [114](#)
- iv\_humidite\_ext
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [114](#)
- iv\_humidite\_int
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [114](#)
- iv\_poids
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [114](#)
- iv\_pression
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
- iv\_tem\_ext
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
- iv\_tem\_int
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
- latitude
  - fr : :campus : :laurainc : :honeybee : :Ruche, [226](#)
- layout\_alertes
  - fr : :campus : :laurainc : :honeybee : :homeActivity, [119](#)
- layout\_debug
  - fr : :campus : :laurainc : :honeybee : :homeActivity, [119](#)
- layout\_gestionAlertes
  - fr : :campus : :laurainc : :honeybee : :homeActivity, [119](#)
- layout\_gestionRuches
  - fr : :campus : :laurainc : :honeybee : :homeActivity, [119](#)
- layout\_graphs
  - fr : :campus : :laurainc : :honeybee : :homeActivity, [119](#)
- layout\_map
  - fr : :campus : :laurainc : :honeybee : :homeActivity, [119](#)
- layout\_tableauDeBord
  - fr : :campus : :laurainc : :honeybee : :homeActivity, [119](#)
- listeChoixAppID
  - fr : :campus : :laurainc : :honeybee : :Ruche, [227](#)
- listeChoixRuches
  - fr : :campus : :laurainc : :honeybee : :Ruche, [227](#)
- listesTTN
  - NouvelleRuchelhm, [184](#)
- longitude
  - fr : :campus : :laurainc : :honeybee : :Ruche, [227](#)
- m\_linearLayout
  - fr : :campus : :laurainc : :honeybee : :MainActivity, [166](#)
- mFormat
  - fr : :campus : :laurainc : :honeybee : :TimeAsXAxisLabel↔  
Formatter, [296](#)
- mMap
  - fr : :campus : :laurainc : :honeybee : :CarteActivity, [65](#)
- mSeriesEnsoleillement
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [227](#)
- mSeriesHumiditeExt
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [227](#)
- mSeriesHumiditeInt
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
- fr : :campus : :laurainc : :honeybee : :Ruche, [227](#)
- mSeriesPoids
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [227](#)
- mSeriesPression
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [228](#)
- mSeriesTemperaturesExt
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [116](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [228](#)
- mSeriesTemperaturesInt
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [116](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [228](#)
- maRuche
  - Ruchelhm, [293](#)
- main
  - main.cpp, [306](#)
- main.cpp, [306](#)
- main, [306](#)
- MainActivity.java, [307](#)
- menuOnglets
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, [179](#)
- mesRuches
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [95](#)
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [115](#)
  - Ruchelhm, [294](#)
- messageConnexion
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees : :↔  
ConnexionMySQL, [85](#)
- messageRecu
  - Communication, [81](#)
- MesureHoraireEnvironnement, [167](#)
  - heure, [168](#)
  - humiditeExterieurMaximum, [168](#)
  - humiditeExterieurMinimum, [168](#)
  - humiditeExterieurMoyenne, [168](#)
  - humiditesRecues, [169](#)
  - pressionAtmospheriqueMaximum, [169](#)
  - pressionAtmospheriqueMinimum, [169](#)
  - pressionAtmospheriqueMoyenne, [169](#)
  - pressionAtmospheriqueRecues, [169](#)
  - temperatureExterieurMaximum, [169](#)
  - temperatureExterieurMinimum, [169](#)
  - temperatureExterieurMoyenne, [170](#)
  - temperaturesRecues, [170](#)
- mesureHoraireEnvironnement
  - Ruche, [262](#)
- MesureHoraireRuche, [170](#)
  - heure, [171](#)
  - humiditeInterieurMaximum, [171](#)
  - humiditeInterieurMinimum, [171](#)
  - humiditeInterieurMoyenne, [171](#)
  - humiditesRecues, [171](#)
  - temperatureInterieurMaximum, [172](#)
  - temperatureInterieurMinimum, [172](#)
  - temperatureInterieurMoyenne, [172](#)
  - temperaturesRecues, [172](#)
- mesureHoraireRuche
  - Ruche, [262](#)
- MesuresHoraireEnsoleillement, [172](#)

- ensoleillementsMaximum, 173
  - ensoleillementsMinimum, 173
  - ensoleillementsMoyen, 173
  - ensoleillementsRecues, 173
  - heure, 173
- mesuresHorraireEnsoleillement
  - Ruche, 262
- mettreAJourListeRuches
  - RucheIhm, 288
- mqttAndroidClient
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, 75
- mutex
  - BaseDeDonnees, 52
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, 63
- nbAcces
  - BaseDeDonnees, 52
- nettoyerZoneDeTexte
  - NouvelleRucheIhm, 183
- nom
  - fr : :campus : :laurainc : :honeybee : :Ruche, 228
- nomRuche
  - Alertes, 38
- nomTopic
  - Communication, 82
- nouvelleDonnesRuche
  - RucheIhm, 288
- nouvelleHumiditeExterieur
  - Ruche, 245
- nouvelleHumiditeInterieur
  - Ruche, 245
- nouvelleMesureCharge
  - Ruche, 245
- nouvelleMesureCourant
  - Ruche, 246
- nouvelleMesureEnsoleillement
  - Ruche, 246
- nouvelleMesureEnsoleillementTraite
  - Ruche, 246
- nouvelleMesureEnsoleillementTraiteParHeure
  - Ruche, 246
- nouvelleMesureHumiditeExterieurTraite
  - Ruche, 246
- nouvelleMesureHumiditeExterieurTraiteParHeure
  - Ruche, 247
- nouvelleMesureHumiditeInterieurTraite
  - Ruche, 247
- nouvelleMesureHumiditeInterieurTraiteParHeure
  - Ruche, 247
- nouvelleMesurePoids
  - Ruche, 247
- nouvelleMesurePressionAtmospheriqueTraiteParHeure
  - Ruche, 247
- nouvelleMesureTemperatureExterieurTraite
  - Ruche, 248
- nouvelleMesureTemperatureExterieurTraiteParHeure
  - Ruche, 248
- nouvelleMesureTemperatureInterieurTraite
  - Ruche, 248
- nouvelleMesureTemperatureInterieurTraiteParHeure
  - Ruche, 248
- nouvelleMesureTension
  - Ruche, 248
- nouvellePressionAtmospherique
  - Ruche, 249
- nouvellePressionAtmospheriqueTraite
  - Ruche, 249
- nouvelleRuche
  - RucheIhm, 294
- NouvelleRucheActivity.java, 307
- NouvelleRucheIhm, 180
  - ~NouvelleRucheIhm, 183
- annulerDonneesNouvelleRuche, 183
- bdd, 184
- listesTTN, 184
- nettoyerZoneDeTexte, 183
- NouvelleRucheIhm, 182
- recevoirDonneeAjoutRuche, 183
- ui, 184
- nouvelleRucheIhm.cpp, 307
- nouvelleRucheIhm.h, 308
- nouvelleTemperatureExterieur
  - Ruche, 249
- nouvelleTemperatureInterieur
  - Ruche, 249
- onActivityResult
  - fr : :campus : :laurainc : :honeybee : :MainActivity, 164
- onCreate
  - fr : :campus : :laurainc : :honeybee : :CarteActivity, 64
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, 91
  - fr : :campus : :laurainc : :honeybee : :DetailsRucheActivity, 97
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 112
  - fr : :campus : :laurainc : :honeybee : :MainActivity, 164
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 176
  - fr : :campus : :laurainc : :honeybee : :ParametresHoneyBeeActivity, 186
  - fr : :campus : :laurainc : :honeybee : :alertesActivity, 42
  - fr : :campus : :laurainc : :honeybee : :homeActivity, 118
- onMapReady
  - fr : :campus : :laurainc : :honeybee : :CarteActivity, 65
- onPostExecute
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees : :↔ ConnexionMySQL, 84
- onProgressUpdate
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees : :↔ ConnexionMySQL, 84
- ouvrir
  - BaseDeDonnees, 47
- ouvrirReglagesAlertes
  - RucheIhm, 289
- PASSWORD\_GMAIL
  - parametres.h, 310
- POIDS\_SEUIL\_MAX
  - parametres.h, 311
- POIDS\_SEUIL\_MIN
  - parametres.h, 311
- PORT\_MESURE\_ENERGIE
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, 76

PORT\_MESURE\_ENSOLEILLEMENT  
     fr : :campus : :laurainc : :honeybee : :ClientMQTT, 76  
 PORT\_MESURE\_ENVIRONNEMENT  
     fr : :campus : :laurainc : :honeybee : :ClientMQTT, 76  
 PORT\_MESURE\_POIDS  
     fr : :campus : :laurainc : :honeybee : :ClientMQTT, 76  
 PORT\_MESURE\_RUCHE  
     fr : :campus : :laurainc : :honeybee : :ClientMQTT, 76  
 PRESSION\_ATMOSPHERIQUE\_SEUIL\_MAX  
     parametres.h, 311  
 PRESSION\_ATMOSPHERIQUE\_SEUIL\_MIN  
     parametres.h, 311  
 parametres.h, 308  
     APP\_TITRE, 309  
     BATTERIE\_SEUIL\_MIN, 309  
     BDD\_NOMBASE, 309  
     BDD\_PASSWORD, 309  
     BDD\_SERVEUR, 309  
     BDD\_USERNAME, 310  
     ENSOLEILLEMENT\_SEUIL\_MAX, 310  
     ENSOLEILLEMENT\_SEUIL\_MIN, 310  
     HUMIDITE\_EXTERIEUR\_SEUIL\_MAX, 310  
     HUMIDITE\_EXTERIEUR\_SEUIL\_MIN, 310  
     HUMIDITE\_INTERIEUR\_SEUIL\_MAX, 310  
     HUMIDITE\_INTERIEUR\_SEUIL\_MIN, 310  
     PASSWORD\_GMAIL, 310  
     POIDS\_SEUIL\_MAX, 311  
     POIDS\_SEUIL\_MIN, 311  
     PRESSION\_ATMOSPHERIQUE\_SEUIL\_MAX, 311  
     PRESSION\_ATMOSPHERIQUE\_SEUIL\_MIN, 311  
     PortsTTN, 312  
     SeuilsAlertes, 313  
     TEMPERATURE\_EXTERIEUR\_SEUIL\_MAX, 311  
     TEMPERATURE\_EXTERIEUR\_SEUIL\_MIN, 311  
     TEMPERATURE\_INTERIEUR\_SEUIL\_MAX, 311  
     TEMPERATURE\_INTERIEUR\_SEUIL\_MIN, 311  
     TTN\_EMAIL, 312  
     TTN\_PASSWORD, 312  
     TTN\_PORT, 312  
     TTN\_SERVEUR, 312  
     TTN\_TOPIC, 312  
     TTN\_USERNAME, 312  
     TypeAlertes, 313  
     USER\_GMAIL, 312  
 ParametresHoneyBeeActivity.java, 314  
 password  
     fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, 63  
     fr : :campus : :laurainc : :honeybee : :ClientMQTT, 75  
     fr : :campus : :laurainc : :honeybee : :MainActivity, 166  
     fr : :campus : :laurainc : :honeybee : :ParametresHoneyBeeActivity, 187  
 poids  
     DonneesRuche, 103  
     fr : :campus : :laurainc : :honeybee : :Ruche, 228  
     InfosPoids, 146  
 poids\_Basse  
     fr : :campus : :laurainc : :honeybee : :Ruche, 228  
 poids\_Haute  
     fr : :campus : :laurainc : :honeybee : :Ruche, 228  
 poids\_Moyenne  
     fr : :campus : :laurainc : :honeybee : :Ruche, 229  
 poids\_afficher  
     fr : :campus : :laurainc : :honeybee : :GraphActivity, 116  
 poidsEnvoye  
     InfosPoids, 145  
 PortsTTN  
     parametres.h, 312  
 portsTTn  
     Communication, 83  
 pression  
     fr : :campus : :laurainc : :honeybee : :Ruche, 229  
 pression\_Basse  
     fr : :campus : :laurainc : :honeybee : :Ruche, 229  
 pression\_Haute  
     fr : :campus : :laurainc : :honeybee : :Ruche, 229  
 pression\_Moyenne  
     fr : :campus : :laurainc : :honeybee : :Ruche, 229  
 pression\_afficher  
     fr : :campus : :laurainc : :honeybee : :GraphActivity, 116  
 pressionAtmospherique  
     DonneesRuche, 103  
     InfosPressionAtmospherique, 151  
 pressionAtmospheriqueEnvoye  
     InfosPressionAtmospherique, 149  
 pressionAtmospheriqueMaximum  
     MesureHoraireEnvironnement, 169  
 pressionAtmospheriqueMinimum  
     MesureHoraireEnvironnement, 169  
 pressionAtmospheriqueMoyenne  
     MesureHoraireEnvironnement, 169  
 pressionAtmospheriqueRecues  
     MesureHoraireEnvironnement, 169  
 pressionAtmospheriqueRuche  
     Ruche, 262  
 pressionAtmospheriques  
     InfosPressionAtmospherique, 152  
 QDialog, 188  
 QObject, 188  
 QWidget, 188  
 quitter  
     Ruchelhm, 289  
 README.md, 314  
 REQUETE\_SQL\_ALERTES  
     fr : :campus : :laurainc : :honeybee : :HoneyBee, 121  
 REQUETE\_SQL\_ERREUR  
     fr : :campus : :laurainc : :honeybee : :HoneyBee, 121  
 REQUETE\_SQL\_IDTTN  
     fr : :campus : :laurainc : :honeybee : :HoneyBee, 121  
 REQUETE\_SQL\_LISTE\_RUCHES  
     fr : :campus : :laurainc : :honeybee : :HoneyBee, 121  
 REQUETE\_SQL\_MESURES\_RUCHES  
     fr : :campus : :laurainc : :honeybee : :HoneyBee, 121  
 REQUETE\_SQL\_OK  
     fr : :campus : :laurainc : :honeybee : :HoneyBee, 122  
 REQUETE\_SQL\_RUCHE  
     fr : :campus : :laurainc : :honeybee : :HoneyBee, 122  
 readResolve  
     fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, 60  
 receptionner  
     Communication, 81

- recevoirAlertesBatterie
  - Ruche, [249](#)
- recevoirAlertesEnsoleillement
  - Ruche, [249](#)
- recevoirAlertesHumiditeExterieur
  - Ruche, [250](#)
- recevoirAlertesHumiditeInterieur
  - Ruche, [250](#)
- recevoirAlertesPoids
  - Ruche, [250](#)
- recevoirAlertesPressionAtmospherique
  - Ruche, [250](#)
- recevoirAlertesTemperatureExterieur
  - Ruche, [251](#)
- recevoirAlertesTemperatureInterieur
  - Ruche, [251](#)
- recevoirDonneeAjoutRuche
  - NouvelleRucheIhm, [183](#)
- recevoirEnsoleillementTraite
  - Ruche, [251](#)
- recevoirEtatClientConnexion
  - Ruche, [252](#)
  - RucheIhm, [289](#)
- recevoirMessage
  - Ruche, [252](#)
- recevoirMesureChargeTraite
  - Ruche, [252](#)
- recevoirMesureCourantTraite
  - Ruche, [253](#)
- recevoirMesureHumiditeExterieurTraite
  - Ruche, [253](#)
- recevoirMesureHumiditeInterieurTraite
  - Ruche, [253](#)
- recevoirMesurePoidsTraite
  - Ruche, [254](#)
- recevoirMesureTensionTraite
  - Ruche, [254](#)
- recevoirPressionAtmospheriqueTraite
  - Ruche, [254](#)
- recevoirReglagesAlertes
  - ReglagesAlertesIhm, [193](#)
- recevoirSeuilBatterie
  - Alertes, [31](#)
- recevoirSeuilEnsoleillement
  - Alertes, [31](#)
- recevoirSeuilHumiditeExterieurMax
  - Alertes, [31](#)
- recevoirSeuilHumiditeExterieurMin
  - Alertes, [32](#)
- recevoirSeuilHumiditeInterieurMax
  - Alertes, [32](#)
- recevoirSeuilHumiditeInterieurMin
  - Alertes, [32](#)
- recevoirSeuilPoidsMax
  - Alertes, [33](#)
- recevoirSeuilPoidsMin
  - Alertes, [33](#)
- recevoirSeuilPressionAtmospherique
  - Alertes, [33](#)
- recevoirSeuilTemperatureExterieurMax
  - Alertes, [34](#)
- recevoirSeuilTemperatureExterieurMin
  - Alertes, [34](#)
- recevoirSeuilTemperatureInterieurMax
  - Alertes, [34](#)
- recevoirSeuilTemperatureInterieurMin
  - Alertes, [35](#)
- recevoirTemperatureExterieurTraite
  - Ruche, [255](#)
- recevoirTemperatureInterieurTraite
  - Ruche, [255](#)
- recevoirTraitementEnsoleillement
  - Ruche, [256](#)
- recevoirTraitementHumiditeExterieur
  - Ruche, [256](#)
- recevoirTraitementHumiditeInterieur
  - Ruche, [256](#)
- recevoirTraitementPressionAtmospherique
  - Ruche, [257](#)
- recevoirTraitementTemperatureExterieur
  - Ruche, [257](#)
- recevoirTraitementTemperatureInterieur
  - Ruche, [258](#)
- reconnecter
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [60](#)
- recuperer
  - BaseDeDonnees, [48–51](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [212](#)
- recupererChoixChApplID
  - fr : :campus : :laurainc : :honeybee : :Ruche, [214](#)
- recupererDonnees
  - Ruche, [258](#)
- recupererElementsIHM
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, [177](#)
- recupererElementsVue
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [91](#)
- recupererHistoriqueAlertes
  - fr : :campus : :laurainc : :honeybee : :Ruche, [214](#)
- recupererHorodatage
  - Ruche, [259](#)
- recupererIdTTN
  - fr : :campus : :laurainc : :honeybee : :Ruche, [215](#)
- recupererListeRuches
  - fr : :campus : :laurainc : :honeybee : :Ruche, [216](#)
- recupererLocalisationPasserelle
  - Ruche, [259](#)
- recupererMesuresJournalieresRuche
  - fr : :campus : :laurainc : :honeybee : :Ruche, [217](#)
- recupererMoyennes
  - fr : :campus : :laurainc : :honeybee : :Ruche, [219](#)
- recupererNumeroDePort
  - Ruche, [260](#)
- ReglagesAlertesIhm, [189](#)
  - ~ReglagesAlertesIhm, [190](#)
  - alertes, [195](#)
  - bdd, [195](#)
  - envoiSeuilEnsoleillementMin, [191](#)
  - envoiSeuilHumiditeExterieurMax, [191](#)
  - envoiSeuilHumiditeExterieurMin, [191](#)
  - envoiSeuilHumiditeInterieurMax, [191](#)
  - envoiSeuilHumiditeInterieurMin, [191](#)



- envoiSeuilPoidsMax, 192
- envoiSeuilPoidsMin, 192
- envoiSeuilPressionAtmospheriqueMin, 192
- envoiSeuilTemperatureExterieurMax, 192
- envoiSeuilTemperatureExterieurMin, 192
- envoiSeuilTemperatureInterieurMax, 192
- envoiSeuilTemperatureInterieurMin, 193
- recevoirReglagesAlertes, 193
- ReglagesAlertesIhm, 190
- setAlertes, 194
- showEvent, 194
- ui, 195
- reglagesAlertesIhm
  - RucheIhm, 294
- reglagesAlertesIhm.cpp, 314
- reglagesAlertesIhm.h, 314
- reinitialiserCourbes
  - RucheIhm, 290
- renitialiserMarqueurs
  - RucheIhm, 290
- reponsePingTTN
  - Communication, 82
- Ruche, 231
  - ~Ruche, 234
  - alertes, 260
  - bdd, 261
  - communicationRuche, 261
  - connecterGestionMessageBrut, 235
  - connecterGestionMessageTraite, 235
  - donneesRuche, 261
  - donneesRucheTTN, 261
  - ensoleillementRuche, 261
  - envoiAlertesBatterie, 236
  - envoiAlertesEnsoleillement, 236
  - envoiAlertesHumiditeExterieur, 236
  - envoiAlertesHumiditeInterieur, 236
  - envoiAlertesPoids, 236
  - envoiAlertesPressionAtmospherique, 237
  - envoiAlertesTemperatureExterieur, 237
  - envoiAlertesTemperatureInterieur, 237
  - envoiLocalisationPasserelle, 237
  - etatClientConnexion, 237
  - extraireDonnees, 237
  - extraireDonneesBrutes, 239
  - extrairePoids, 239
  - fr : :campus : :laurainc : :honeybee : :Ruche, 198, 199
  - gererAlertes, 240
  - getAlertes, 240
  - getIdRuche, 241
  - humiditeRuche, 261
  - infosBatterie, 262
  - infosPoids, 262
  - insérerDonneesPortBatterie, 241
  - insérerDonneesPortEnsoleillement, 241
  - insérerDonneesPortMesureEnvironnement, 242
  - insérerDonneesPortMesureRuche, 242
  - insérerDonneesPortPoids, 242
  - insérerMesureHoraireBatterie, 243
  - insérerMesureHoraireEnsoleillement, 243
  - insérerMesureHoraireEnvironnement, 243
  - insérerMesureHorairePoids, 244
  - insérerMesureHoraireRuche, 244
  - mesureHoraireEnvironnement, 262
  - mesureHoraireRuche, 262
  - mesuresHorraireEnsoleillement, 262
  - nouvelleHumiditeExterieur, 245
  - nouvelleHumiditeInterieur, 245
  - nouvelleMesureCharge, 245
  - nouvelleMesureCourant, 246
  - nouvelleMesureEnsoleillement, 246
  - nouvelleMesureEnsoleillementTraite, 246
  - nouvelleMesureEnsoleillementTraiteParHeure, 246
  - nouvelleMesureHumiditeExterieurTraite, 246
  - nouvelleMesureHumiditeExterieurTraiteParHeure, 247
  - nouvelleMesureHumiditeInterieurTraite, 247
  - nouvelleMesureHumiditeInterieurTraiteParHeure, 247
  - nouvelleMesurePoids, 247
  - nouvelleMesurePressionAtmospheriqueTraiteParHeure, 247
  - nouvelleMesureTemperatureExterieurTraite, 248
  - nouvelleMesureTemperatureExterieurTraiteParHeure, 248
  - nouvelleMesureTemperatureInterieurTraite, 248
  - nouvelleMesureTemperatureInterieurTraiteParHeure, 248
  - nouvelleMesureTension, 248
  - nouvellePressionAtmospherique, 249
  - nouvellePressionAtmospheriqueTraite, 249
  - nouvelleTemperatureExterieur, 249
  - nouvelleTemperatureInterieur, 249
  - pressionAtmospheriqueRuche, 262
  - recevoirAlertesBatterie, 249
  - recevoirAlertesEnsoleillement, 249
  - recevoirAlertesHumiditeExterieur, 250
  - recevoirAlertesHumiditeInterieur, 250
  - recevoirAlertesPoids, 250
  - recevoirAlertesPressionAtmospherique, 250
  - recevoirAlertesTemperatureExterieur, 251
  - recevoirAlertesTemperatureInterieur, 251
  - recevoirEnsoleillementTraite, 251
  - recevoirEtatClientConnexion, 252
  - recevoirMessage, 252
  - recevoirMesureChargeTraite, 252
  - recevoirMesureCourantTraite, 253
  - recevoirMesureHumiditeExterieurTraite, 253
  - recevoirMesureHumiditeInterieurTraite, 253
  - recevoirMesurePoidsTraite, 254
  - recevoirMesureTensionTraite, 254
  - recevoirPressionAtmospheriqueTraite, 254
  - recevoirTemperatureExterieurTraite, 255
  - recevoirTemperatureInterieurTraite, 255
  - recevoirTraitementEnsoleillement, 256
  - recevoirTraitementHumiditeExterieur, 256
  - recevoirTraitementHumiditeInterieur, 256
  - recevoirTraitementPressionAtmospherique, 257
  - recevoirTraitementTemperatureExterieur, 257
  - recevoirTraitementTemperatureInterieur, 258
  - recupererDonnees, 258
  - recupererHorodatage, 259
  - recupererLocalisationPasserelle, 259
  - recupererNumeroDePort, 260
  - Ruche, 233
  - temperatureRuche, 263
- ruce
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, 116

- fr : :campus : :laurainc : :honeybee : :alertesActivity, 42
- ruche.cpp, 315
- ruche.h, 316
- Ruche.java, 316
- Ruchelhm, 263
  - ~Ruchelhm, 266
  - afficherAlertesBatterie, 266
  - afficherAlertesEnsoleillement, 267
  - afficherAlertesHumiditeExterieur, 268
  - afficherAlertesHumiditeInterieur, 269
  - afficherAlertesPoids, 269
  - afficherAlertesPressionAtmospherique, 270
  - afficherAlertesTemperatureExterieur, 270
  - afficherAlertesTemperatureInterieur, 271
  - afficherBatterie, 272
  - afficherEnsoleillement, 272
  - afficherHumiditeExterieur, 272
  - afficherHumiditeInterieur, 273
  - afficherLocalisationPasserelle, 273
  - afficherMesuresJournalieresEnsoleillement, 273
  - afficherMesuresJournalieresEnvironnement, 274
  - afficherMesuresJournalieresRuche, 274
  - afficherMesuresJournalieresSelectionnee, 275
  - afficherPoids, 277
  - afficherPressionAtmospherique, 277
  - afficherTemperatureExterieur, 277
  - afficherTemperatureInterieur, 278
- bdd, 292
- courbeEnsoleillement, 293
- courbeHumiditeExterieur, 293
- courbeHumiditeInterieur, 293
- courbePressionAtmospherique, 293
- courbeTemperatureExterieur, 293
- courbeTemperatureInterieur, 293
- creerNouvelleRuche, 278
- deconnecterSignaux, 278
- effacerTableau, 279
- initialiserEnsoleillementCourbe, 280
- initialiserHumiditeExterieurCourbe, 280
- initialiserHumiditeInterieurCourbe, 280
- initialiserMarqueurAlerteEnsoleillement, 281
- initialiserMarqueurAlerteHumiditeExterieur, 281
- initialiserMarqueurAlerteHumiditeInterieur, 281
- initialiserMarqueurAlertePression, 282
- initialiserMarqueurAlerteTemperatureExterieur, 282
- InitialiserMarqueurAlerteTemperatureInterieur, 282
- initialiserPressionAtmospheriqueCourbe, 283
- initialiserTemperatureExterieurCourbe, 283
- initialiserTemperatureInterieurCourbe, 283
- initialiserThermometre, 284
- initialiserWidgets, 284
- initialiserCourbes, 285
- insérerMesuresEnsoleillementTableau, 286
- insérerMesuresEnvironnementTableau, 287
- insérerMesuresRucheTableau, 287
- maRuche, 293
- mesRuches, 294
- mettreAJourListeRuches, 288
- nouvelleDonnesRuche, 288
- nouvelleRuche, 294
- ouvrirReglagesAlertes, 289
- quitter, 289
- recevoirEtatClientConnexion, 289
- reglagesAlertesIhm, 294
- reinitialiserCourbes, 290
- renitialiserMarqueurs, 290
- Ruchelhm, 265
- selectionnerRuche, 290
- supprimerRuche, 292
- ui, 294
- ruchelhm.cpp, 317
- ruchelhm.h, 317
- rucheUtilitaire
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, 180
- selectionnerRuche
  - Ruchelhm, 290
- serverUri
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, 76
- setActivite
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, 61
- setAlertes
  - ReglagesAlertesIhm, 194
- setCallback
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, 74
- setCharge
  - fr : :campus : :laurainc : :honeybee : :Ruche, 221
- setEnsoleillement
  - fr : :campus : :laurainc : :honeybee : :Ruche, 221
  - InfosEnsoleillement, 131
- setHandler
  - fr : :campus : :laurainc : :honeybee : :Ruche, 221
- setHorodatage
  - fr : :campus : :laurainc : :honeybee : :Ruche, 222
- setHorodatageBatterie
  - InfosBatterie, 125
- setHorodatageEnsoleillement
  - InfosEnsoleillement, 131
- setHorodatageHumiditeExterieur
  - InfosHumidite, 138
- setHorodatageHumiditeInterieur
  - InfosHumidite, 139
- setHorodatagePoids
  - InfosPoids, 145
- setHorodatagePressionAtmospherique
  - InfosPressionAtmospherique, 150
- setHorodatageTemperatureExterieur
  - InfosTemperature, 157
- setHorodatageTemperatureInterieur
  - InfosTemperature, 158
- setHumiditeExt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 222
- setHumiditeExterieur
  - InfosHumidite, 139
- setHumiditeInt
  - fr : :campus : :laurainc : :honeybee : :Ruche, 222
- setHumiditeInterieur
  - InfosHumidite, 139
- setIdRuche
  - fr : :campus : :laurainc : :honeybee : :DetailsRucheActivity, 98



- setInfosBatterie
  - Alertes, [35](#)
- setInfosEnsoleillement
  - Alertes, [35](#)
- setInfosHumidite
  - Alertes, [35](#)
- setInfosPoids
  - Alertes, [36](#)
- setInfosPressionAtmospherique
  - Alertes, [36](#)
- setInfosTemperature
  - Alertes, [36](#)
- setMesuresBatterie
  - InfosBatterie, [125](#)
- setParametres
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [61](#)
- setPoids
  - fr : :campus : :laurainc : :honeybee : :Ruche, [222](#)
  - InfosPoids, [145](#)
- setPression
  - fr : :campus : :laurainc : :honeybee : :Ruche, [222](#)
- setPressionAtmospherique
  - InfosPressionAtmospherique, [150](#)
- setSubscriptionTopic
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [74](#)
- setTemperatureExt
  - fr : :campus : :laurainc : :honeybee : :Ruche, [223](#)
- setTemperatureExterieur
  - InfosTemperature, [158](#)
- setTemperatureInt
  - fr : :campus : :laurainc : :honeybee : :Ruche, [223](#)
- setTemperatureInterieur
  - InfosTemperature, [158](#)
- seuilBatterie
  - Alertes, [38](#)
- seuilEnsoleillement
  - Alertes, [38](#)
- seuilHumiditeExterieurMax
  - Alertes, [38](#)
- seuilHumiditeExterieurMin
  - Alertes, [38](#)
- seuilHumiditeInterieurMax
  - Alertes, [38](#)
- seuilHumiditeInterieurMin
  - Alertes, [39](#)
- seuilPoidsMax
  - Alertes, [39](#)
- seuilPoidsMin
  - Alertes, [39](#)
- seuilPressionAtmospherique
  - Alertes, [39](#)
- seuilTemperatureExterieurMax
  - Alertes, [39](#)
- seuilTemperatureExterieurMin
  - Alertes, [39](#)
- seuilTemperatureInterieurMax
  - Alertes, [40](#)
- seuilTemperatureInterieurMin
  - Alertes, [40](#)
- SeuilsAlertes
  - parametres.h, [313](#)
- seuilsAlertes
  - Alertes, [39](#)
- showEvent
  - ReglagesAlertesIhm, [194](#)
- souscrireTopic
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [74](#)
- subscriptionTopic
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [76](#)
- supprimerRuche
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [62](#)
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [92](#)
  - Ruchelhm, [292](#)
- TAG
  - fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [63](#)
  - fr : :campus : :laurainc : :honeybee : :ClientMQTT, [77](#)
  - fr : :campus : :laurainc : :honeybee : :DashboardActivity, [95](#)
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [116](#)
  - fr : :campus : :laurainc : :honeybee : :MainActivity, [166](#)
  - fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity, [180](#)
  - fr : :campus : :laurainc : :honeybee : :ParametresHoneyBeeActivity, [187](#)
  - fr : :campus : :laurainc : :honeybee : :Ruche, [229](#)
  - fr : :campus : :laurainc : :honeybee : :alertesActivity, [42](#)
- TEMPERATURE\_EXTERIEUR\_SEUIL\_MAX
  - parametres.h, [311](#)
- TEMPERATURE\_EXTERIEUR\_SEUIL\_MIN
  - parametres.h, [311](#)
- TEMPERATURE\_INTERIEUR\_SEUIL\_MAX
  - parametres.h, [311](#)
- TEMPERATURE\_INTERIEUR\_SEUIL\_MIN
  - parametres.h, [311](#)
- TTN\_EMAIL
  - parametres.h, [312](#)
- TTN\_PASSWORD
  - parametres.h, [312](#)
- TTN\_PORT
  - parametres.h, [312](#)
- TTN\_SERVEUR
  - parametres.h, [312](#)
- TTN\_TOPIC
  - parametres.h, [312](#)
- TTN\_USERNAME
  - parametres.h, [312](#)
- temp\_ext\_Basse
  - fr : :campus : :laurainc : :honeybee : :Ruche, [229](#)
- temp\_ext\_Haute
  - fr : :campus : :laurainc : :honeybee : :Ruche, [230](#)
- temp\_ext\_Moyenne
  - fr : :campus : :laurainc : :honeybee : :Ruche, [230](#)
- temp\_ext\_afficher
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [116](#)
- temp\_int\_Basse
  - fr : :campus : :laurainc : :honeybee : :Ruche, [230](#)
- temp\_int\_Haute
  - fr : :campus : :laurainc : :honeybee : :Ruche, [230](#)
- temp\_int\_Moyenne
  - fr : :campus : :laurainc : :honeybee : :Ruche, [230](#)
- temp\_int\_afficher
  - fr : :campus : :laurainc : :honeybee : :GraphActivity, [116](#)

temperatureExt  
     fr : :campus : :laurainc : :honeybee : :Ruche, 230  
 temperatureExterieur  
     DonneesRuche, 103  
     InfosTemperature, 161  
 temperatureExterieurEnvoye  
     InfosTemperature, 158  
 temperatureExterieurMaximum  
     MesureHoraireEnvironnement, 169  
 temperatureExterieurMinimum  
     MesureHoraireEnvironnement, 169  
 temperatureExterieurMoyenne  
     MesureHoraireEnvironnement, 170  
 temperatureInt  
     fr : :campus : :laurainc : :honeybee : :Ruche, 230  
 temperatureInterieur  
     DonneesRuche, 103  
     InfosTemperature, 162  
 temperatureInterieurEnvoye  
     InfosTemperature, 159  
 temperatureInterieurMaximum  
     MesureHoraireRuche, 172  
 temperatureInterieurMinimum  
     MesureHoraireRuche, 172  
 temperatureInterieurMoyenne  
     MesureHoraireRuche, 172  
 temperatureRuche  
     Ruche, 263  
 temperaturesExterieurs  
     InfosTemperature, 162  
 temperaturesInterieurs  
     InfosTemperature, 162  
 temperaturesRecues  
     MesureHoraireEnvironnement, 170  
     MesureHoraireRuche, 172  
 temps\_reel\_afficher  
     fr : :campus : :laurainc : :honeybee : :GraphActivity, 116  
 tension  
     DonneesBatterie, 99  
     InfosBatterie, 127  
 tensionEnvoye  
     InfosBatterie, 125  
 TimeAsXAxisLabelFormatter  
     fr : :campus : :laurainc : :honeybee : :TimeAsXAxisLabel←  
         Formatter, 295  
 TimeAsXAxisLabelFormatter.java, 318  
 traitementEnsoleillementEnvoye  
     InfosEnsoleillement, 131  
 traitementHumiditeExterieurEnvoye  
     InfosHumidite, 140  
 traitementHumiditeInterieurEnvoye  
     InfosHumidite, 140  
 traitementPressionAtmospheriqueEnvoye  
     InfosPressionAtmospherique, 150  
 traitementTemperatureExterieurEnvoye  
     InfosTemperature, 159  
 traitementTemperatureInterieurEnvoye  
     InfosTemperature, 159  
 traiterMessage  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 92  
 traiterMoyenneHoraireTemperatureInterieur  
     InfosTemperature, 159  
 traiterNouveauPoids  
     InfosPoids, 146  
 traiterNouvelleEnsoleillement  
     InfosEnsoleillement, 132  
 traiterNouvelleHumiditeExterieur  
     InfosHumidite, 140  
 traiterNouvelleHumiditeInterieur  
     InfosHumidite, 141  
 traiterNouvelleMesureCharge  
     InfosBatterie, 126  
 traiterNouvelleMesureCourant  
     InfosBatterie, 126  
 traiterNouvelleMesureTension  
     InfosBatterie, 126  
 traiterNouvellePressionAtmospherique  
     InfosPressionAtmospherique, 150  
 traiterNouvelleTemperatureExterieur  
     InfosTemperature, 160  
 traiterNouvelleTemperatureInterieur  
     InfosTemperature, 160  
 tv\_date  
     fr : :campus : :laurainc : :honeybee : :NouvelleRucheActivity,  
         180  
 tv\_donneesBas  
     fr : :campus : :laurainc : :honeybee : :GraphActivity, 117  
 tv\_donneesHaut  
     fr : :campus : :laurainc : :honeybee : :GraphActivity, 117  
 tv\_donneesMoyenne  
     fr : :campus : :laurainc : :honeybee : :GraphActivity, 117  
 tv\_erreur  
     fr : :campus : :laurainc : :honeybee : :MainActivity, 166  
 tv\_historiqueAlertes  
     fr : :campus : :laurainc : :honeybee : :alertesActivity, 43  
 tv\_valeurCourante  
     fr : :campus : :laurainc : :honeybee : :GraphActivity, 117  
 txtAlertes  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 95  
 txtCharge  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtDescription  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtEnsoleillement  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtGPS  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtHorodatage  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtHumiditeExt  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtHumiditeInt  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtPoids  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtPression  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 96  
 txtTemperatureExt  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 97  
 txtTemperatureInt  
     fr : :campus : :laurainc : :honeybee : :DashboardActivity, 97  
 TypeAlertes

parametres.h, [313](#)

typeBase

- BaseDeDonnees, [52](#)

USER\_GMAIL

- parametres.h, [312](#)

Ui, [16](#)

ui

- NouvelleRuchelhm, [184](#)
- ReglagesAlertesIhm, [195](#)
- Ruchelhm, [294](#)

url

- fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [63](#)
- fr : :campus : :laurainc : :honeybee : :MainActivity, [167](#)

useAppContext

- fr : :campus : :laurainc : :honeybee : :Example↔  
InstrumentedTest, [104](#)

username

- fr : :campus : :laurainc : :honeybee : :BaseDeDonnees, [63](#)
- fr : :campus : :laurainc : :honeybee : :ClientMQTT, [77](#)
- fr : :campus : :laurainc : :honeybee : :MainActivity, [167](#)
- fr : :campus : :laurainc : :honeybee : :ParametresHoney↔  
BeeActivity, [187](#)

valider

- fr : :campus : :laurainc : :honeybee : :ParametresHoney↔  
BeeActivity, [186](#)