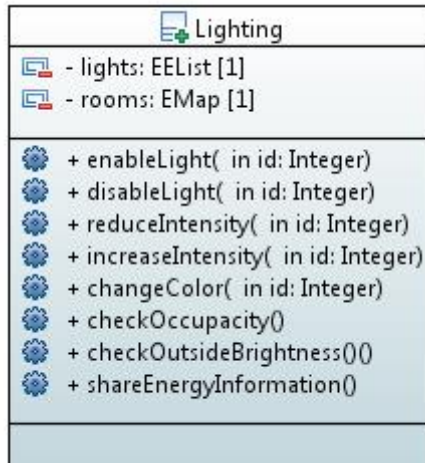


## Définitions des méthodes

### I. Ligthing



### Propriétés

lights	Liste des différentes Lumières du bâtiments. Chaque Lumières étant un objet « Light » possédant les attributs : <ul style="list-style-type: none"><li>• is_on (indiquant si elle est allumée)</li><li>• intensity (indiquant son intensité)</li><li>• color (indiquant sa couleur)</li><li>• time (indiquant son temps d'utilisation)</li></ul>
rooms	Map de type <Integer, List<Light>>, la clé étant l'id de la room et la valeur étant la liste des lumières qui lui sont associé.

### Méthodes

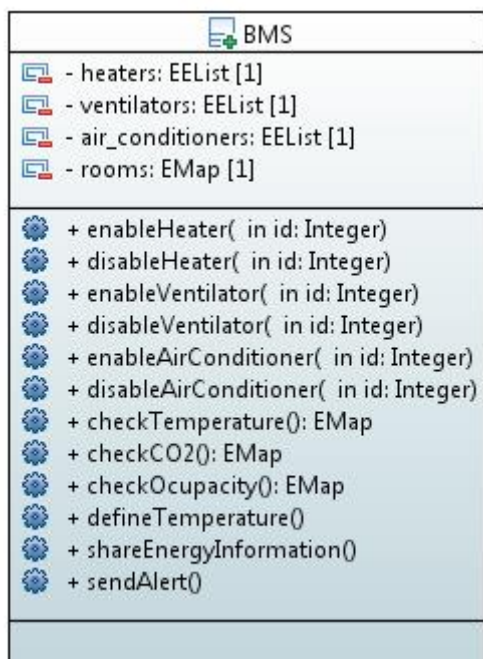
enableLight(id : Integer)	Allume la lumière correspondant à l'id.
disableLight(id : Integer)	Eteins la lumière correspondant à l'id.
reduceIntensity(id : Integer)	Réduit l'intensité de la lumière correspondant à l'id.
increaseIntensity(id : Integer)	Augmente l'intensité de la lumière correspondant à l'id.
changeColor(id : Integer)	Change la couleur de la lumière correspondant à l'id.
checkOccupacy()	Contrôle l'occupation de toutes les rooms du bâtiment à l'aide de détecteur de mouvement. Pour chaque room : <ul style="list-style-type: none"><li>• S'il y a occupation et que les lumières sont éteintes, alors appel de la méthode enableLight() sur les lumières associées à la room.</li></ul>

- S'il n'y a pas occupation et que les lumières sont allumées, alors appel de la méthode `disableLight()` sur les lumières associées à la room.
- Sinon rien.

<code>checkOutsideBrightness()</code>	Contrôle la luminosité extérieure, adaptant ainsi de manière cohérente l'intensité de toutes les lumières du bâtiments grâce aux méthodes <code>reduceIntensity()</code> et <code>increaseIntensity()</code> .
<code>shareEnergyInformation()</code>	Partage sur une base de données commune aux différents CS ses données de consommation énergétique. Ici, elles sont liées au temps de mise en marche des lumières.

**Remarque :** Les méthodes `checkOccupacity()` et `checkOutsideBrightness()` sont appelé automatiquement à un intervalle définis au préalable.

## II. BMS



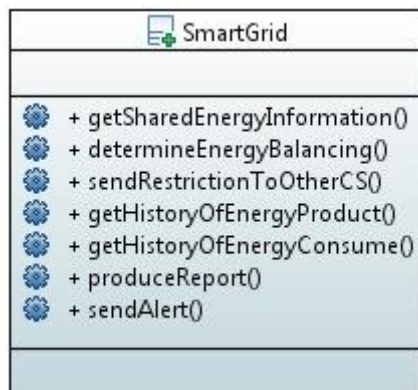
### Propriétés

<code>heaters</code>	Liste des différents radiateurs du bâtiment.
<code>ventilators</code>	Liste des différents ventilateurs du bâtiment.
<code>air_contiditionners</code>	Liste des différents conditionneurs d'air du bâtiment.
<code>rooms</code>	Map de type <code>&lt;Integer, List&lt;Heater Ventilator AirConditioner&gt;&gt;</code> , la clé étant l'id de la room et la valeur étant la liste des radiateurs, ventilateur et conditionneur d'air qui lui sont associé.

## Méthodes

<code>enableHeater(id : Integer)</code>	Allume le radiateur correspondant à l'id.
<code>disableHeater(id : Integer)</code>	Eteins le radiateur correspondant à l'id.
<code>enableVentilator(id : Integer)</code>	Allume le ventilateur correspondant à l'id.
<code>disableVentilator(id : Integer)</code>	Eteins le ventilateur correspondant à l'id.
<code>enableAirConditioner(id : Integer)</code>	Allume le conditionneur d'air correspondant à l'id.
<code>disableAirConditioner(id : Integer)</code>	Eteins le conditionneur d'air correspondant à l'id.
<code>checkTemperature() : EMap</code>	Contrôle la température de toutes les rooms et renvoie une Map de type <Integer, Integer>, la clé étant l'id de la room et la valeur étant la température de la room.
<code>checkCO2() : EMap</code>	Contrôle le niveau de CO2 de toutes les rooms et renvoie une Map de type <Integer, Integer>, la clé étant l'id de la room et la valeur étant le niveau de CO2 de la room.
<code>checkOccupancy() : EMap</code>	Contrôle l'occupation de toutes les rooms et renvoie une Map de type <Integer, Boolean>, la clé étant l'id de la room et la valeur étant un Boolean représentant l'occupation de la room.
<code>defineTemperature()</code>	<p>A partir de :</p> <ul style="list-style-type: none"><li>• La température extérieure</li><li>• Le résultat de <code>checkTemperature()</code></li><li>• Le résultat de <code>checkCO2()</code></li><li>• <code>checkOccupancy()</code></li></ul> <p>La méthode va pouvoir allumer, éteindre ou bien gérer la puissance des différents appareils du bâtiment (radiateur, ventilateur, conditionneur d'air).</p>
<code>shareEnergyInformation()</code>	Partage sur une base de données commune aux différents CS ses données de consommation énergétique. Ici, elles sont liées au temps de mise en marche des différents appareils du bâtiment (radiateur, ventilateur, conditionneur d'air) ainsi que de la puissance sur laquelle ils ont été positionnés.
<code>sendAlert()</code>	Envoie une alerte au CS <i>fire detection</i> si <code>checkTemperature()</code> détecte une température anormalement élevé.

### III. Smart Grid



#### Propriétés

Aucune.

#### Méthodes

<code>getSharedEnergyInformation()</code>	Obtiens les informations énergétiques partagées par les différents CS.
<code>determineEnergyBalancing()</code>	Détermine un ratio entre l'énergie produite et l'énergie consommée. Si ce ratio se traduit par une trop grosse dépense en énergie par rapport à la production, alors appel de la méthode <code>sendRestrictionToOtherCS()</code> .
<code>sendRestrictionToOtherCS()</code>	Appeler si problèmes de balance énergétique. Envoie les restrictions nécessaires au(x) CS(s) responsable de cette perte énergétique. (ex : peux appeler la méthode <code>reduceIntensity()</code> sur toutes les lumières d'une room consommant de manière excessive)
<code>getHistoryOfEnergyProduct()</code>	Obtiens l'historique de la production d'énergie.
<code>getHistoryOfEnergyConsume()</code>	Obtiens l'historique de la consommation d'énergie .
<code>produceReport()</code>	Produit un rapport automatiquement rempli par rapport aux historiques de production/consommation énergétique. Met en lumières les facteurs de grosses dépenses énergétique ou à l'inverse les facteurs de grosses productions énergétique.
<code>sendAlert()</code>	Envoie des alertes énergétiques au CS de gestion des urgences en cas de panne de courant.