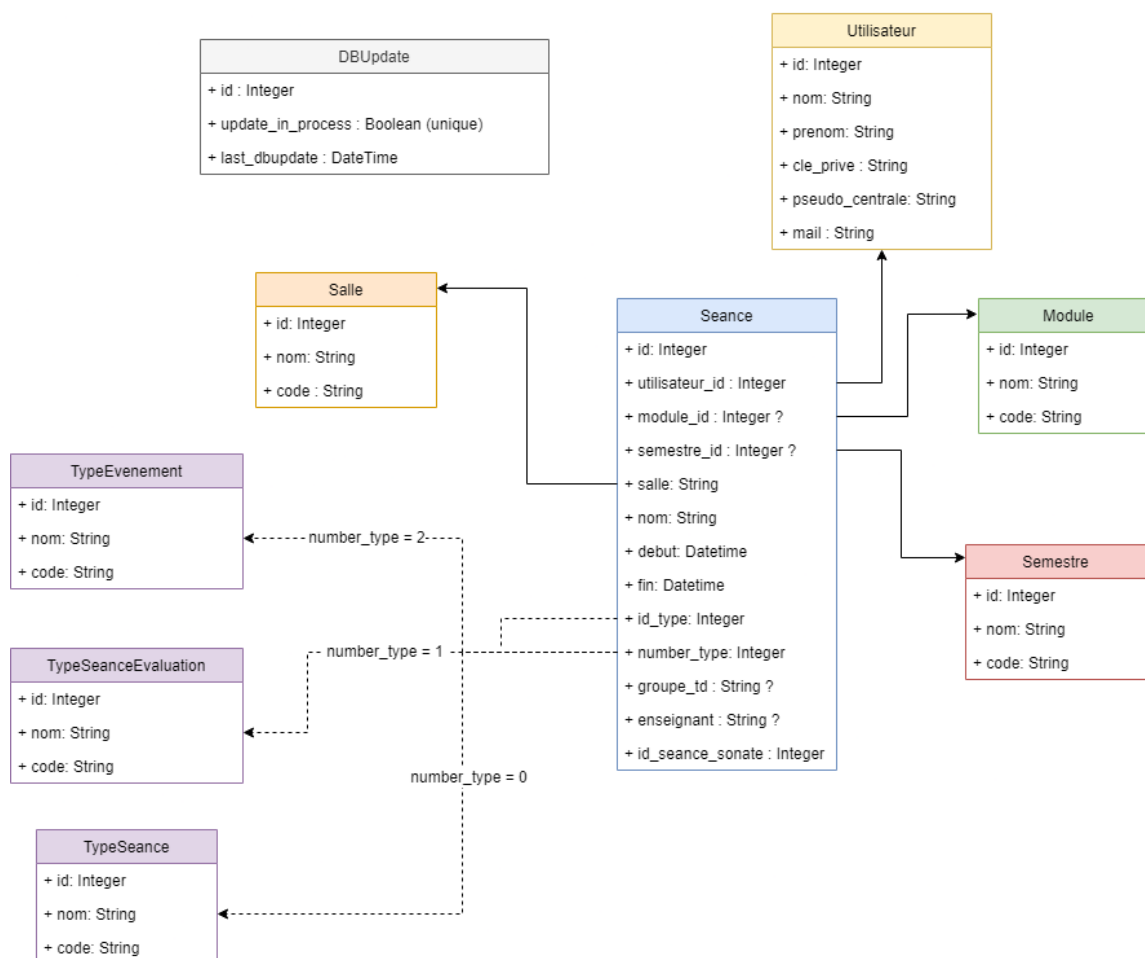


# Sérénade - GInfo



## Gestion des salles

Afin de gérer la possibilité d'avoir plusieurs salles sur une séance, le champ `salle` de la table `Seance` est un string. Nous le remplissons avec `id1|id2|id3|...` avec `id1`, `id2`, `id3`, ... les id des différentes salles.

**Exemple** : Une séance avec `1|4|12` dans le champ `salle` se déroule dans la salle avec l'id `1`, la salle avec l'id `4` et la salle avec l'id `12`.

## Gestion des enseignants

Afin de gérer la possibilité d'avoir plusieurs enseignants sur une séance, le champ `Enseignant` de la table `Seance` est un string géré de la même façon que le champ `salle`.

Comme il est possible qu'une séance n'ait pas d'enseignant ou que l'enseignant ne soit pas encore connu, nous procédons de cette façon :

- `enseignant` : `null` → cette séance n'a et n'aura jamais d'enseignant (exemple: Travail en Autonomie)
- `enseignant` : `""` (chaîne de caractère vide) → l'enseignant est encore à déterminer. Nous affichons "Enseignant : À déterminer"

**Exemple** : Une séance avec `1|4|12` dans le champ `enseignant` se déroule avec l'enseignant ayant l'id `1`, l'enseignant ayant l'id `4` et l'enseignant ayant l'id `12`.

## Gestion des types de séance

Afin de gérer les différents types de séance de Sonate, nous avons besoin des trois tables `TypeSeance`, `TypeSeanceEvaluation` et `TypeEvenement`. Ainsi dans la table `Seance` nous avons le champ `number_type` qui vaut `0`, `1` ou `2` et nous indique ainsi s'il faut utiliser respectivement la table `TypeSeance`, `TypeSeanceEvaluation` et `TypeEvenement`. Ensuite le champ `id_type` nous indique l'id du Type a utilisé dans la table déterminée par `number_type`.

**Exemple** : Une séance avec `2` dans le champ `number_type` et `14` dans le champ `id_type` fait ainsi référence au type ayant l'id `14` dans la table `TypeEvenement`.

Le couple (`id_seance_sonate`, `number_type`) est unique et permet de réunir les personnes présentes à la même séance.

## Point sur les semestres

Nous avons eu précédemment le souci que lorsqu'une personne regarde l'emploi du temps d'une autre personne ayant un homonyme, l'emploi du temps affiché contient les séances des deux homonymes. Nous avons géré ce cas en utilisant le pseudo CAS plutôt que le nom et prénom pour afficher l'emploi du temps d'une autre personne que la personne connectée. Bien que le pseudo CAS ne soit pas une clé primaire en théorie, cela devrait réduire le nombre de fois où ce genre de problèmes se produit. Cependant, il nous paraît pertinent de continuer à afficher les semestres (par exemple dans le cas où un élève aurait les cours de plusieurs semestres).

Ainsi le semestre est géré via un champ `semestre` dans la table `Seance` qui fait référence à la table `Semestre`. Dans l'emploi du temps nous afficherons le code du `semestre` qui sera "S5" par exemple.

## Point sur la table DBUpdate

La structure de la table reste inchangée par rapport à l'ancienne version de Sérénade.

## Aspect Technique

- Le site, réalisé en Symfony 5.4, utilise PHP 8.1, Composer 2.3 et Node 16.
- La base de données sera en PostgreSQL13.
- Le script d'import se fait toutes les 2 heures (comme avant).