

Petites Annonces Cité Universitaire : Purge des annonces

1. Introduction.

Cette documentation donne un début de solution suite à la réflexion d'améliorer l'application de location de la future cité universitaire. En effet, les modules utilisateurs et d'administration sont déjà opérationnels. Cependant, on peut encore améliorer l'application en automatisant la purge de la base de données des annonces.

2. Arborescence du projet.

Dans l'image suivante, on observe l'arborescence du « Projet-annonces » qui contient l'application en elle même ainsi que son module d'administration.

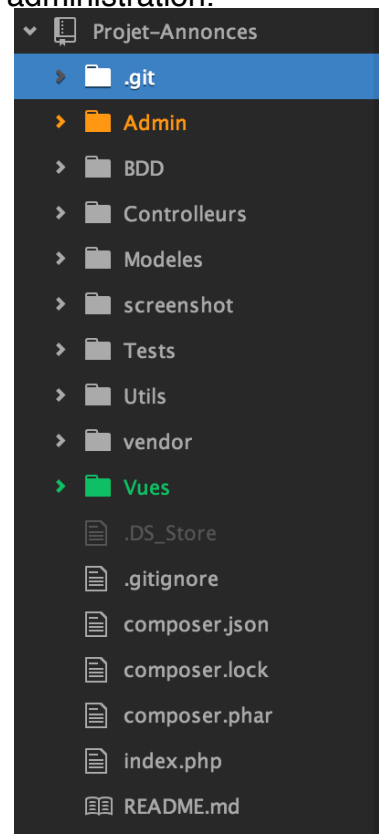
Le dossier « [Admin](#) » comporte les vues et les traitements dédiés au module d'admin du projet.

Le dossier « BDD » comporte un fichier « [annonce.sql](#) » permettant de peupler votre pour pour pouvoir utiliser l'application.

Les contrôleurs placés dans le fichier « [Controleurs](#) » implémente des méthodes associées respectivement à un modèle du dossier « [Modeles](#) ».

Le dossier « [Utils](#) » comporte des fichiers inclus (par le biais) dans vos vues et traitement afin de pouvoir invoquer toutes les entités du projet.

Le dossier « [Vues](#) » comporte toutes les pages de présentation, les scripts js et ressources bootstrap du projet.



3. Présentation du statut d'une annonce :

Les constantes sont dotées d'un attribut statut. Ce dernier peut prendre trois états; chacun associer à une valeur numérique :

états	Valeur numérique associée
NON_TRAITE	10
TRAITE	20
A_SUPPRIMER	30

Remarque : L'association d'un état à une valeur numérique a été mise en place afin de simplifier les traitements tout en gardant une bonne lisibilité du code.

Petites Annonces Cité Universitaire : Purge des annonces

4. Scénario d'automatisation de la purge des annonces.

L'automatisation de la purge de la base de données repose sur la mise en place d'une durée de vie des annonces. L'état d'une annonce changera alors suite aux manipulations de l'administrateur (Cette fonctionnalité est déjà implémentée) d'une part et pour une certaine durée de vie donnée d'autre part.

Une fois l'annonce validée par l'administrateur du site, un mail de mise en garde sera envoyé, au bout d'une semaine, à l'émetteur de l'annonce afin de lui demander s'il souhaite reconduire son annonce. Si l'utilisateur ne répond pas au mail sous *48 heures* ; son annonce sera supprimée.

Conclusion :

- Une annonce à une durée de vie de *96 heures* à partir du moment où elle a été validée par un administrateur ou par un script automatique.
- Un mail d'avertissement doit être envoyé toutes les semaines.
- Pour une durée de vie d'une annonce comprise entre *24 et 48 heures (après la limite d'une semaine)*, il faut être capable de prendre en considération son retour mail.

5. Mise en place.

Remarque préliminaire : Ce paragraphe expose une voie pour mettre en place le module d'automatisation de la purge de la table « `annonceM` ».

a. Manipulation des entités du projet.

La purge de la table « `annonceM` » repose sur l'évaluation de la durée de vie de l'annonce.

Pour ce cela, ajouter un premier champ « `$dureeDeVie` » à la classe « `AnnonceM.php` » et une colonne « `DUREE_DE_VIE` » de type « `TIMESTAMP` » à la table « `annonceM` ». Initialiser l'attribut « `$dureeDeVie` » à « `null` » à dans le constructeur de la classe.

Ajouter un deuxième champ « `$mail` » à la classe « `AnnonceM.php` » et une colonne « `MAIL` » de type « `BOOLEAN` » à la table « `annonceM` ». Initialiser l'attribut « `$mail` » à « `true` » à dans le constructeur de la classe.

Petites Annonces Cité Universitaire : Purge des annonces

b. Initialisation de la durée de vie d'une annonce.

Lorsque l'annonce est validée par un administrateur, son statut est modifié dans le fichier de traitement « [validation-annonce-treatment.php](#) ». C'est alors qu'il faut initialiser le champ « [\\$dureeDeVie](#) ».

```
if($_validation === true){  
    $_annonceM->statut      = TRAITE;  
    $_annonceM->dureeDeVie = time();  
}
```

c. Purge automatique de la base de données.

Pour exécuter une tâche à intervalle de temps régulier, il faut configurer le serveur de sorte qu'il exécute un script PHP à intervalle de temps régulier. Le paramétrage consiste à indiquer au serveur :

- l'intervalle de temps de au bout duquel il faut exécuter le script à nouveau.
- l'url du script à exécuter d'automatisation.

Remarque : L'intervalle de temps doit être choisi judicieusement pour des d'actualisation des données et de performance du serveur.

d. Contenu du script.

Le script (que l'on pourra nommé « [automatic-annonce-updating.php](#) » par exemple) comportera 3 parties majeures de traitement.

La première partie consistera en l'évaluation de la durée de vie de chacune des annonces présentes dans la base de données.

```
$_annonces = $_annonceC->getAllAnnonces();  
foreach($_annonces as $_annonceM){  
    $_start = $_annonce->dureeDeVie;  
    $_end   = time();  
    $_annonce->dureeDeVie = $_end - $_start;  
    $_annonceC->updateAnnonce($_annonceM);  
}
```

Petites Annonces Cité Universitaire : Purge des annonces

Attention : On peut s'inspirer sur code précédant mais il ne correspond pas forcément à l'algorithme final.

La seconde partie portera sur la mise à jour de l'attribut « `$mail` » et de l'envoi éventuel d'un mail de mise en garde à un utilisateur.

```
$o_annonceC = new AnnonceC();

$o_serviceMail = new ServiceMail() // developper un service mail

$_annonces = $o_annonceC->getAllAnnonces();

foreach( $_annonces as $o_annonceM ){

    $_envoyerMail = $o_annonceM->mail;

    $_dureeDeVie = $o_annonceM->dureeDeVie;

    // 172800 = 48h et 345600s = 96h
    if( $_envoyerMail == true && $_dureeDeVie >= 172800 && $_dureeDeVie < 345600 ) {

        $o_annonceM->mail = false;

        // envoie un mail d'avertissement à un utilisateur pour une annonce donné
        $o_serviceMail->envoyeMailAvertissment( $o_annonceM->id, $o_annonceM->idUtilisateur );

    }

}
```

Attention : On peut s'inspirer sur code précédant mais il ne correspond pas forcément à l'algorithme final.

Une dernière partie portera sur la mise à jour de l'attribut « `$statut` » et de la suppression éventuelle de l'annonce d'un utilisateur.

```
$o_annonceC = new AnnonceC();

$o_serviceMail = new ServiceMail() // developper un service mail

$_annonces = $o_annonceC->getAllAnnonces();

foreach( $_annonces as $o_annonceM ){

    $_envoyerMail = $o_annonceM->mail;

    $_dureeDeVie = $o_annonceM->dureeDeVie;

    // 345600s = 96h
    if( $_envoyerMail == false && $_dureeDeVie >= 345600 ) {

        // envoyer un mail de suppression à l'utilisateur
        $o_serviceMail->envoyeMailSuppression( $o_annonceM->id, $o_annonceM->idUtilisateur );

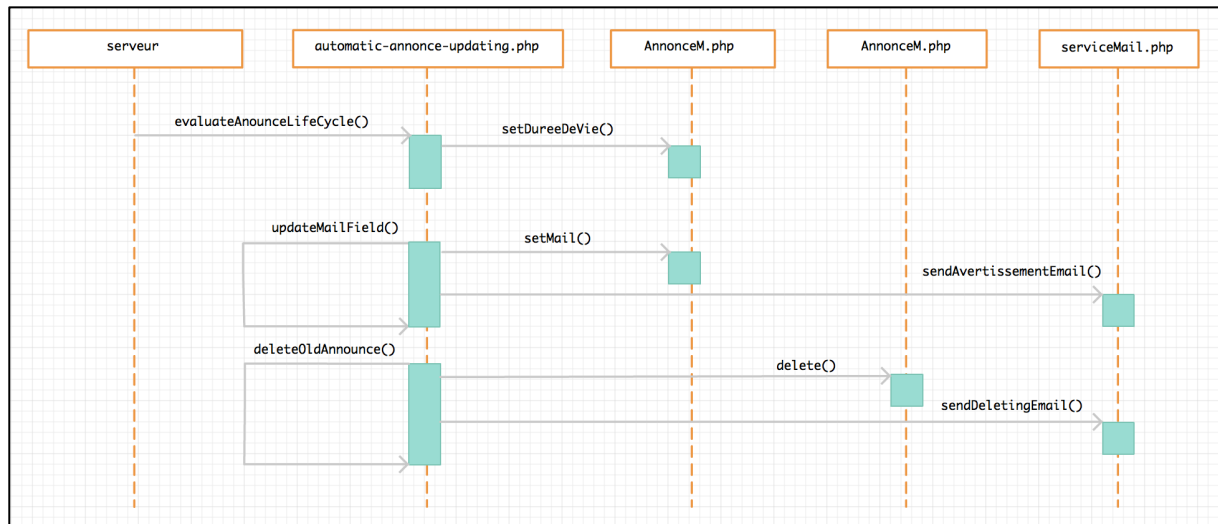
    }

}
```

Petites Annonces Cité Universitaire : Purge des annonces

e. Diagramme de séquence.

Le diagramme de séquence suivant récapitule la démarche pour purger la base de données des annonces grâce à une tâche CRON envoyée depuis le serveur :



6. Prendre en compte la réponse mail d'un utilisateur.

a. Introduction.

Avec la démarche décrite dans la partie précédente. On ne peut pas contrôler la volonté ou non de l'utilisateur à maintenir l'annonce sur le site. Il va falloir alors implémenter une nouvelle chaîne de traitement afin de satisfaire à cette exigence.

b. Principe d'implémentation.

Pour que le module de purge automatique (dédié à la table des annonces) soit complètement fonctionnel. Il faut pouvoir prendre en compte la réponse mail d'un utilisateur. Pour cela, je vous rassure... rien de plus simple !

Il faut exploiter le mail envoyer à l'utilisateur pour l'informer que son annonce sera prochainement supprimer. Insérer, par exemple, dans votre email un bouton lier à l'URL suivante :

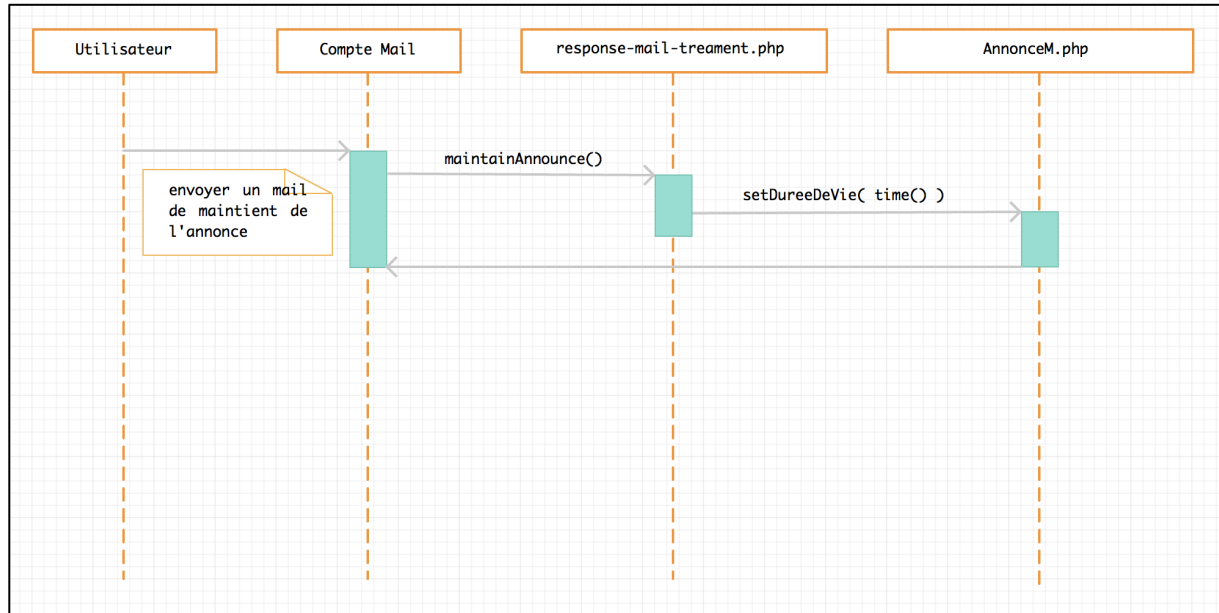
[https://www.\[domain\].fr/Projet-annonces/Admin/reponse-mail-treatment?id=\[idAnnonce\]](https://www.[domain].fr/Projet-annonces/Admin/reponse-mail-treatment?id=[idAnnonce])

On peut alors exploiter le paramètre « `idAnnonce` » dans le script PHP « `response-mail-treatment.php` » pour initialiser à nouveau le champ « `$dureeDeVie` » par la date courante et en remettant le « `$mail` » à « `true` ».

Petites Annonces Cité Universitaire : Purge des annonces

c. Diagramme de séquence.

Le diagramme de séquence suivant résume les interactions entre les différentes entités du projet pour prendre en compte la volonté d'un utilisateur de maintenir son annonce sur le site.



7. Conclusion.

En espérant que ce que ce document vous ait aidé à mieux comprendre l'architecture du projet et à entrevoir la conception de l'automatisation du module de purge de la BDD. L'équipe vous souhaite bon courage pour la l'achèvement du projet.