

**COMPTE RENDU**

Outils de gestion écologique

**Explications technique de la réalisation du logiciel**

Date de la dernière mise à jour : 03 juillet 2020

**Sommaire**

**1. Ecriture du code**

1.1 Choix du langage

**2.  Structure du programme**

2.1 Exemples de codes du programme

[2.1.1 Constructeur pour stocker les informations liées à un événement](https://docs.google.com/document/d/1FMpbOG0mhtLb2BiLzOELjBapfEWpYW8OvsY5lD_EhUI/edit#heading=h.gez3b9q3ar5r)

2.1.2 Page d’affichage du formulaire création d’événements

2.1.3 Enregistrement du formulaire en BDD

2.1.4 Page qui affiche le header

2.1.5 Page qui affiche supprime un événement

[**3. Difficultés**](https://docs.google.com/document/d/1FMpbOG0mhtLb2BiLzOELjBapfEWpYW8OvsY5lD_EhUI/edit#heading=h.z5b9q09iak6a)

**4. Evolutions possibles**

1. Ecriture du code

1.1 Choix du langage

Notre choix s’est orienté sur le Java car c’est le langage que nous “maîtrisons” le mieux.  
Après de multiples recherches, nous avons constaté que la solution pour gérer à la fois des pages Web dynamiques et de la base de donnée SQL serait Java EE.

Suite à un abandon de cette solution que nous n’avons pas réussi à faire fonctionner (voir ci-dessous “Difficultés”), nous sommes partis sur la solution utilisée dans le cours NFA011, à savoir des pages JSP.

A MODIFIER : nous nous sommes aperçu que la méthode utilisée en NFA011 (JSP+SQL), est une solution en mode « bricolage ». Nous avons rencontré de nombreuses difficultés d’échanges entre les pages et notamment les liens relatifs qui ont un fonctionnement aléatoire (quand on fait un rafraichissement de page 9 fois sur 10 la page s’affiche correctement, sinon il y a un message d’erreur).

2 Structure du programme

**PassionBio2** :La racine du projet (2 pour V2)

**src** : pages java  
 **com.cnam.bdd** : toutes les opérations sur la base de donnée.  
 - **connexionBDD.jsp** : connexion à la BDD  
 - **enregistrer\_equipement.jsp** : requête SQL qui enregistre les équipements  
 - **enregistrer\_evenement.jsp** : requête SQL qui enregistre les événements  
 - **lire\_bdd.jsp**: lit les informations (événement + équipement)

**com.cnam.contructeurs** : pages des constructeurs  
 - **equipement.jsp** : Constructeur équipements  
 - **evenement.jsp** : Constructeur événements  
 **com.cnam.crud** : lecture et écriture des données dans la BDD  
 - **modifierEve.jsp** : modification des événements   
 - **supprimerEve.jsp** : suppression des événements

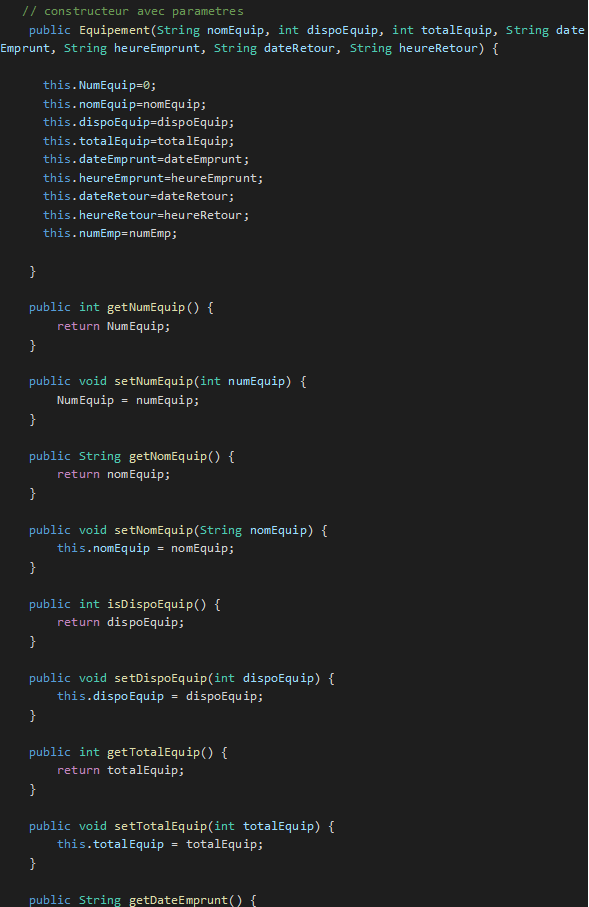
**Webcontent** : pages web  
 **com.cnam.affichages** : affichage des informations  
 - **afficheListeEquipement.jsp** : affiche sous forme de tableau les événements.  
 - **afficheListeEvenement.jsp** : affiche sous forme de tableau les équipements  
 **com.cnam.creation** : saisie des informations  
 - **creation\_equipement.jsp** : formulaire de création d’équipement  
 - **creation\_evenement.jsp** : formulaire de création d’événements.

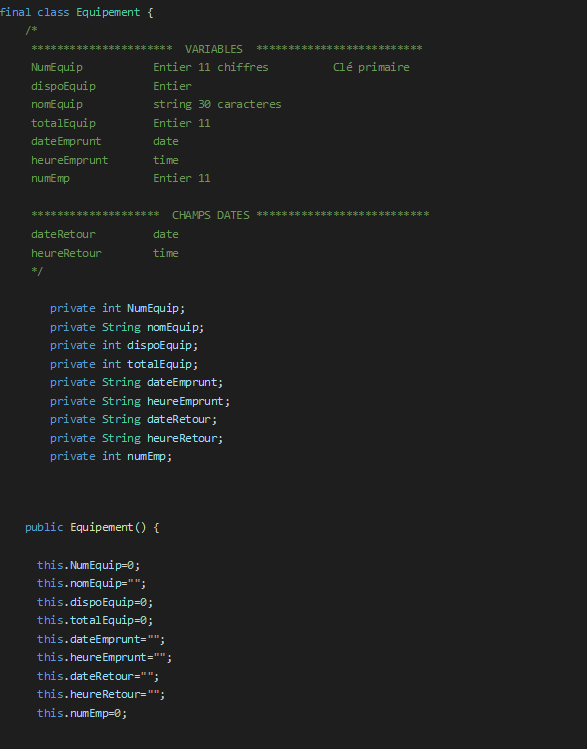
**accueil.jsp** : page d’accueil

A MODIFIER : par manque de temps, nous n’avons pas pu réorganiser la structure.  
il aurait fallu mettre :

* Enregistrer\_equipement.jsp 🡪 dans com.cnam.crud.
* Enregistrer\_evenement.jsp 🡪 dans com.cnam.crud
* Lire\_bdd 🡪 dans com.cnam.crud

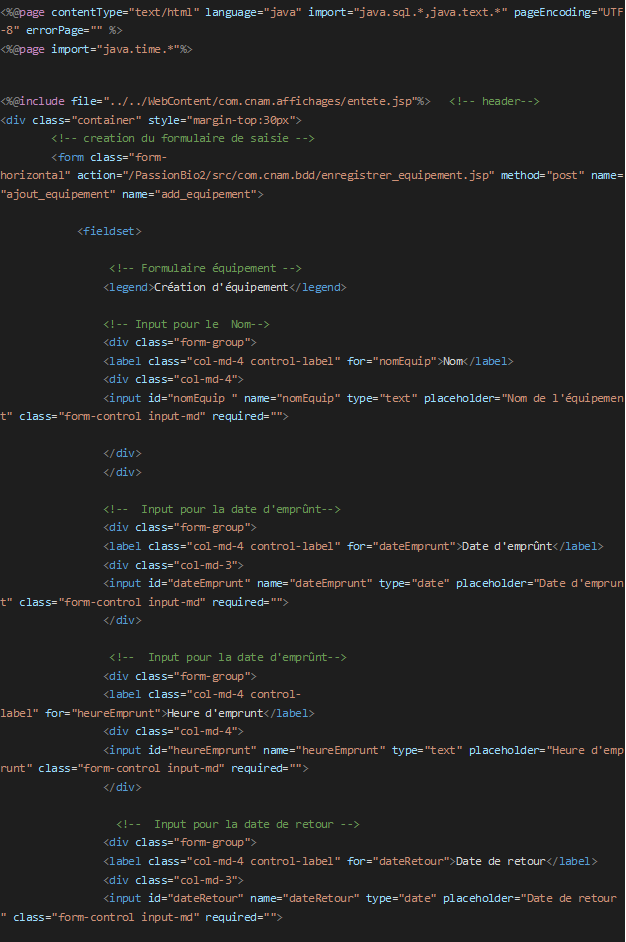
2.1 Exemples de codes du programme

2.1.1 Constructeur pour stocker les informations liées à un événement

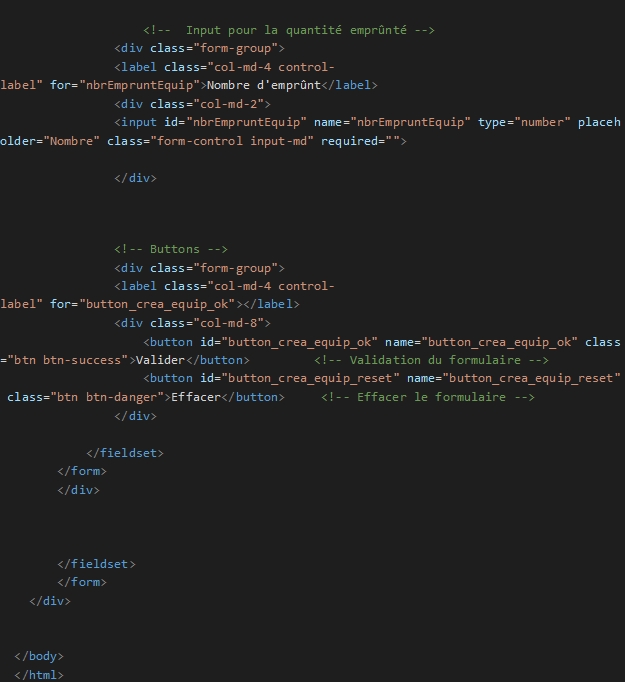


2.1.2 Page d’affichage du formulaire création d’événements.

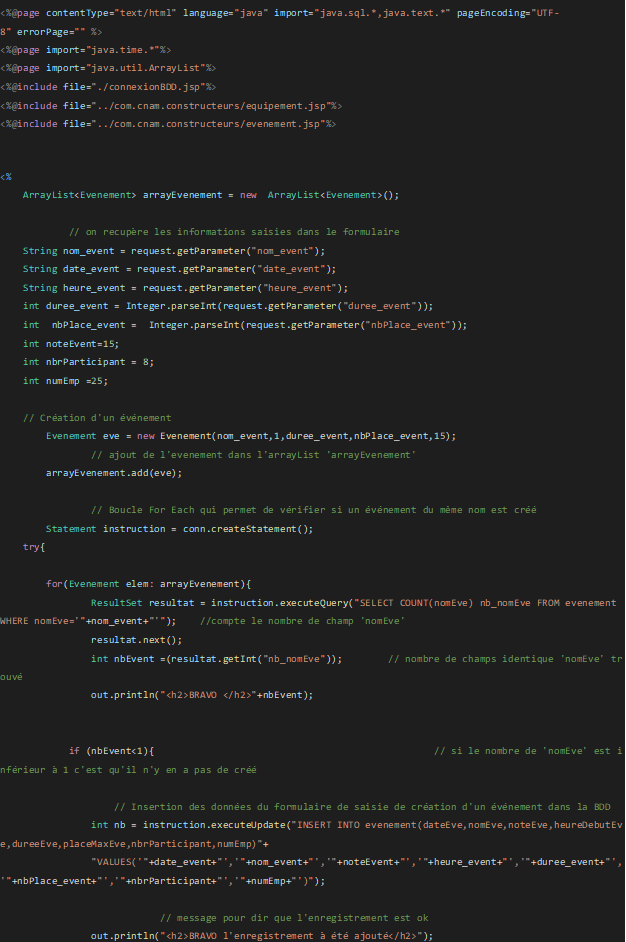
Cette page permet la saisie des informations de la création d’un événement.  
Au clic sur le bouton « valider » le programme renvoi vers la page « enregistrer\_evenement.jsp » qui enregistre ces données dans la BDD en SQL.



A AMELIORER : Validation des champs. Il faut vérifier que chaque champ soit conforme avant d’envoyer les informations en BDD, par exemple, vérifier que la date de réservation de soit pas antérieure à aujourd’hui car on ne peut pas réserver du matériel avant aujourd’hui.



2.1.3 Enregistrement du formulaire en BDD

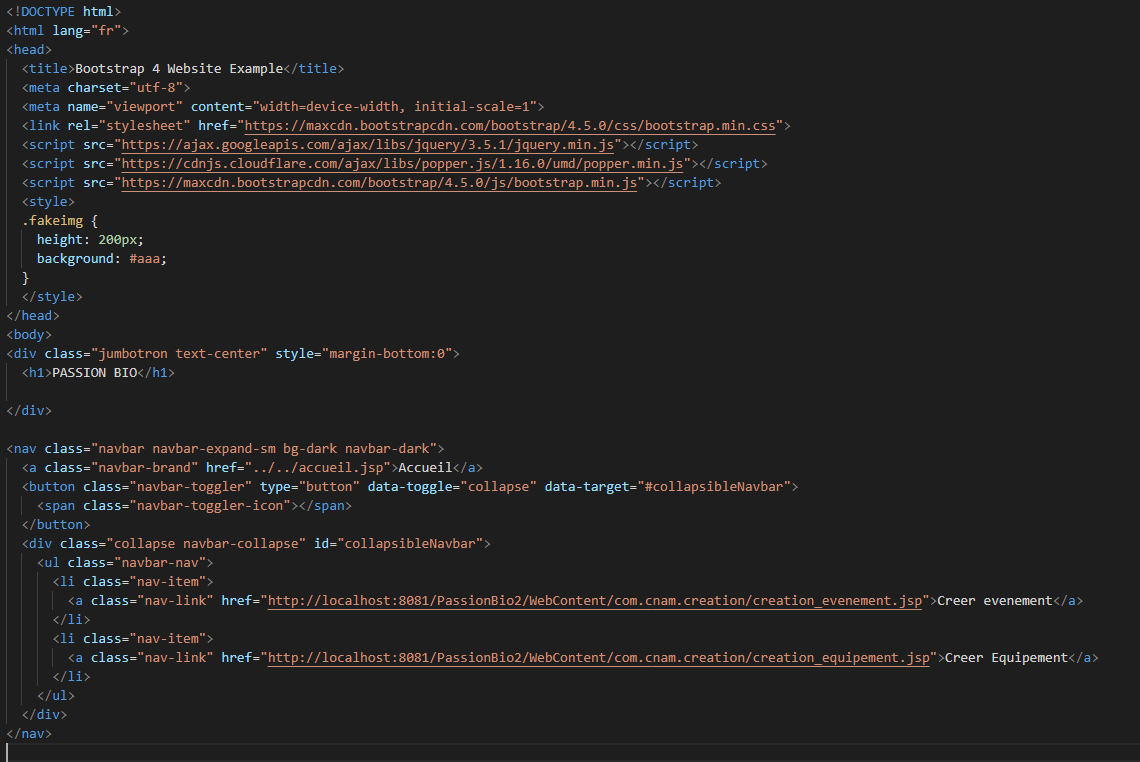


2.1.4 Page qui affiche le header

Ce fichier ’entete\_jsp’ affiche le header qui apparait sur chaque page.  
Il fait apparaitre le nom de la société ‘PassionBio’ et le menu de navigation.

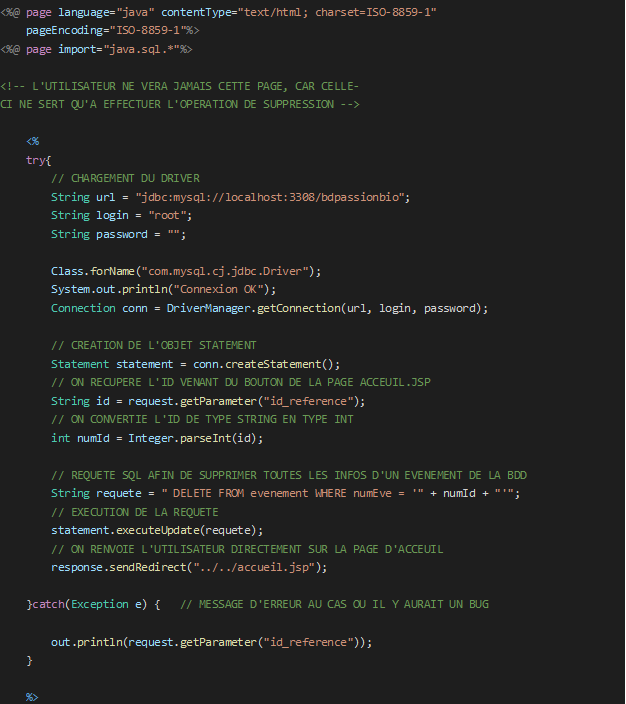
Le but est que chaque page HTML du site fasse appel à cette page afin de ne pas répéter le code à chaque page et faciliter la modification. Si on veut ajouter un menu, la modification se fait une seule fois dans ’entete\_jsp’.





2.1.4 Page qui affiche supprime un événement

Cette page supprime l’évènement sélectionné par l’utilisateur dans le liste des événements.  
 Le programme récupère l’ID de l’événement quand l’utilisateur clique sur le bouton ‘supprimer’ puis grâce à une commande SQL supprime l’enregistrement



AMELIORATIONS : Mettre un pop-up pour demander si l’utilisateur veut vraiment supprimer l’enregistrement afin d’éviter une suppression par inadvertance.

Plutôt que de supprimer définitivement l’enregistrement, il aurait été plus judicieux d’ajouter un champs ‘eveDel’ qui passerai soit à 0 (événement supprimé) ou 1(événement actif). Ceci permettrait de garder un historique dans la base de données qui peut servir ultérieurement, comme par exemple pour des statistiques.

4. Difficultés

Java EE : La phase d’apprentissage a été un peu longue car il n’est pas évident de comprendre le fonctionnement.

Après avoir développé toute la partie de la gestion de connexion des utilisateurs, nous avons butés sur l’établissement de la connexion à la base de donnée. Même avec de l'assistance des professeurs, nous n’y sommes pas arrivés.  
Refonte du projet : Nous avons dû abandonner cette voie pour partir sur une autre solution.

Git : Afin de partager et garder un historique de notre travail, nous avons choisi Git, mais nous avons eu des problèmes liés à l’utilisation de cet outil qui nécessite une formation. De ce fait, des commande git ont engendré des problème de versions.

5. Evolutions possibles

On pourrait ajouter une interface d’administration (permissions) qui permettrai de gérer les droits d’accès en création, modification, suppression d’événement et d’équipements.  
Mais aussi la possibilité de créer des groupes afin de proposer des services uniquement à ceux-ci, comme par exemple, faire une événement uniquement avec la direction.

Développer une version pour les smartphone.  
Avoir un bilan des bénéfices, sur le plan écologique, apporté par les animations et service apportés.  
Personnalisation de l’interface avec des choix de couleurs ou de format de présentations.