

# Documentation Recherche Patch

Application: DBS Patches

Namespace: App\Http\Controllers

Class PatchController extends Controller

## Fonction main:

**-public function Recherche(Request \$request)**

return view avec les patchs et les paramètres de la recherche

*// fonction principale qui va appeler les autres fonctions*

## Fonctions de recherches:

**-public static function RechercheVersionId(\$patchId, \$patchVersion)**

return \$patchs

*//Condition if selon si les deux champs sont remplis ou non puis requete where  
//like pour les versions et where = pour les idPatches*

*// Retourne une collection de patchs.*

**-public static function RechercheMotsCles (\$patchMotsCles)**

return \$patchs

*// on separe la chaîne de caractère avec le separateur espace.*

*// on va faire un where like %.\$motcle.% sur tous les libelles de tags , les libelles  
// de patchs et les commentaires des contenus*

*// Pour ne pas avoir de doublons dans les patchs à chaque requête le résultat est  
passé*

*// dans la fonction statique TableauSansDoublons*

*// On considere que si le champ motcle n'est pas rempli on ne renvoie aucun patch  
// à cause du like %.\$motcle.% on se retrouverait avec tous les patches sinon.*

**-public static function RechercheCodeJiraSalesForce (\$codeJira,  
\$codeSalesForce)**

return \$patches

*// Même principe que pour la fonction RechercheVersionId  
// Seule différence il n'y a pas de requête where like mais uniquement where =*

## **Fonction de trie de tableaux:**

**-public static function TableauSansDoublon (\$patch1,\$patches)**

return \$patches

*// \$patch1 est le tableau à trier , \$patches contiendra aucun doublons .*

**-public static function OccurenceCommune (\$patch1,\$patch2 , \$patch3,\$patches)**

return \$patches

*// \$patch1 et \$patch2 et \$patch3 sont les tableaux à trier ,  
// \$patches contiendra toutes les occurrences communes à ces trois tableaux .*

**-public static function OccurenceCommune2 (\$patch1,\$patch2 , \$patches)**

return \$patches

*// \$patch1 et \$patch2 sont les tableaux à trier ,  
// \$patches contiendra toutes les occurrences communes à ces deux tableaux .*

## **Fonction Word/Excel:**

### **-public static function PatchWord (\$patch)**

pas de return (procédure)

*// dans \public va être généré le word du patch avec pour nom Patch{idPatch}.docx  
// Le code est long à cause de la mise en forme (font , paragraph, saut de ligne).*

### **-public static function PatchExcel (\$patch)**

pas de return (procédure)

*// dans \public va être généré l'excel du patch avec pour nom Patch{idPatch}.xlsx*

## **Détail Fonction main:**

### **-public function Recherche(Request \$request)**

return view avec les patches et les paramètres de la recherche

Appel des fonctions :

- RechercheVersionId(\$patchId, \$patchVersion)
- RechercheMotsCles(\$patchMotsCles)
- RechercheCodeJiraSalesForce(\$codeJira, \$codeSalesforce)

Avec un jeu de condition on regarde quelle recherche de patches est null selon le cas on appelle la fonction d'occurrence commune adéquate.

Si il n'y a qu'une recherche de patches qui n'est pas null on vérifie qu'il n'y ait pas de doublons grâce à la fonction TableauSansDoublon.

Ensuite pour chaque patch de notre recherche on appelle les fonctions PatchWord/PatchExcel qui vont générer un word, un excel pour le patch en paramètre.