***ROME Joachim***

***16/06/2023***

**Note modification et amélioration application – Gestion des représentations**

Lien site intranet :<https://hrepelutest01.intranet.cg43.fr/>

Table des matières

[I – Analyse / concept du code 2](#_Toc137830024)

[II – Amélioration de la structure php / modifications 5](#_Toc137830025)

[**III – Explications de la démarche réalisée** 12](#_Toc137830026)

[IV – Veille technologique, idées d’amélioration 14](#_Toc137830027)

[V – Conclusion 15](#_Toc137830028)

# I – Analyse / concept du code

Création de la page Export\_instance.php (ratacher a 2 fichiers, l’un dans action et l’autre sur le dossier /template)





Toutes les modifications ont étaient effectuer sur le service SSH du repertoire « respresentation-elus-dev », car la version OLD n’est plus actualisé sur le site applicatif.

Les requêtes SQL me renvois les informations que l’on souhaitera afficher plus tard sur la table d’instance.



Par la suite, j’ai du prendre en charge les différents problèmes d’indexation des données présentes sur la base MySQL « Gestion\_repres » depuis MobaXtrem

Des colonnes de sélection d’information étaient alors programmer sans ressortir de champs ou d’informations déjà présentes dans les variables et stocker sous forme objet. D’où l’utilisation de la fonction utf8\_encode, qui est utilisée pour traiter les données avant de les insérer dans la base de données. Cette fonction convertit une chaîne de caractères en une séquence de caractères encodée en UTF-8.

Le rôle de l'encodage UTF-8 est de représenter les caractères Unicode, qui est un standard de codage de caractères prenant en charge un large éventail de langues et de symboles. En utilisant l'encodage UTF-8, les informations stockés seront prisent alors en charge avec la version actuel de PHP 8.2.5 utiliser par le service d’application.

Dans le contexte de la page PHP que j’ai fournie, la fonction utf8\_encode est appliquée aux valeurs des variables $\_POST avant d'insérer ces valeurs dans la base de données.



Ensuite, il est de gérer l'export des instances, probablement à partir d'une base de données.

Voici les principales étapes et fonctionnalités du code :

* Vérification des droits (commenté) : Il y a une ligne commentée qui vérifie les droits de l'utilisateur, mais elle est actuellement désactivée.
* Définition du template et du titre : Le code définit le template à utiliser pour la page d'export et spécifie également le titre de la page.
* Connexion à la base de données : Le code se connecte à une base de données MySQL à l'aide de la fonction connectMYSQLDB(). Cela suggère que les données à exporter sont stockées dans une base de données MySQL.
* Récupération des données d'export : Le code utilise la fonction getExport\_instance() pour récupérer une liste de désignations d'instances à exporter. Il stocke cette liste dans la variable $listDesignations.

#  II – Amélioration de la structure php / modifications

De nombreux bugs d’affichage et d’accès a certaines informations (du a un ajout de fonctionnalité et un passage a une version supérieur de PHP, des adaptations ont été nécessaire)

Correction de champs vides en PHP sans renvoyer aux valeurs enregistrées dans les variables à l'aide des méthodes GET/Try, suivre les étapes suivantes :

 Récupération des données existantes :

 S’Assurer que les données à modifier sont préalablement stockées dans des variables ou récupérées à partir d'une source de données (par exemple, une base de données).

 Vérification des données soumises :

 Lorsque on reçoit les données soumises via un formulaire, effectuer des validations pour s’assurer que les champs nécessaires ne sont pas vides.

 utiliser des conditions et des fonctions de validation pour vérifier chaque champ individuellement.

 Si des champs vides sont détectés, afficher un message d'erreur à l'utilisateur pour l'informer de la correction requise.

 Modification des données :

 Si toutes les données soumises sont valides, procéder à la modification des valeurs existantes.

 Utiliser les nouvelles données soumises pour mettre à jour les variables ou effectuer les modifications dans la source de données (par exemple, la base de données).

utiliser les valeurs correctes lors de la modification et de conserver l'intégrité des autres données associées.

 Notification de la modification :

 Après avoir effectué la modification, informer l'utilisateur que les changements ont été enregistrés avec succès.

 Afficher un message de confirmation ou rediriger l'utilisateur vers une page de confirmation.

Concernant la création d'un script de "drag and drop" pour affecter des élus représentés par des images PNG à une instance en déplaçant leurs images dans un groupe via un script JavaScript, voici une procédure détaillée :

 Préparation de l'interface :

 Crée une interface utilisateur contenant les images PNG représentant les élus et l'instance cible.

 S’assurer d'avoir un conteneur pour chaque groupe et d'attribuer des identifiants uniques à chaque élément image.

 Définition des événements de glisser-déposer :

 Utilisation du gestionnaires d'événements JavaScript pour capturer les actions de glisser-déposer.

 Ajout du gestionnaire d'événements "dragstart" aux images des élus pour marquer l'image en cours de déplacement.

 "dragover" pour autoriser le dépôt d'éléments.

 "drop" pour effectuer l'affectation de l'élu à l'instance.

 Implémentation des fonctions de manipulation d'événements :

 Dans la fonction de gestionnaire d'événements "dragstart", définir les données de transfert en utilisant l'API DataTransfer pour identifier l'élément en cours de déplacement.

 Dans la fonction de gestionnaire d'événements "dragover", empêcher le comportement par défaut pour permettre le dépôt d'éléments.

 Dans la fonction de gestionnaire d'événements "drop", récupérer les données de transfert pour identifier l'élue en cours de déplacement.

 Effectuer les actions nécessaires pour associer l'élue à l'instance cible (par exemple, mettre à jour les données dans la source de données ou effectuer des appels AJAX pour enregistrer les changements).

 Actualisation de l'interface utilisateur :

 Après avoir effectué l'affectation, mettre à jour l'interface utilisateur pour refléter les changements (par exemple, en déplaçant l'image de l'élue dans le groupe correspondant).

Code avant modification :



Voici le code modifié pour l’ajout d’un nouveau représentant depuis la base de données REPRESENTATION (creation\_repres.tpl.php)



Et le résultat obtenu :



Page « Export\_Instance\_tpl.php »



Cette page web (acccèsible depuis l’onglet **Administration**) permet d'exporter les données d'une instance vers un fichier Excel. Voici les explications sur les différentes fonctionnalités et variables :

 Structure HTML : La page est divisée en différentes sections (<div>) avec des classes CSS pour la mise en forme. Il y a une en-tête avec le titre de la page, un formulaire pour l'export, et une section de pied de page.

 Tableau des données d'instance : Le tableau affiche les informations générales sur l'instance, telles que le nom de l'élu, le prénom de l'élu, le numéro de l'instance, le libellé de l'instance, le téléphone et l'adresse e-mail. Les données sont générées dynamiquement à partir d'un tableau $tableData à l'aide d'une boucle foreach.

 Fonction sanitizeInput : Cette fonction est utilisée pour nettoyer les entrées des utilisateurs en utilisant htmlspecialchars et trim. Elle est appelée pour chaque donnée affichée dans le tableau des données pour éviter les problèmes de sécurité.

 Export vers Excel : Lorsque l'utilisateur clique sur le bouton "Exporter vers Excel", le formulaire est soumis et le code PHP correspondant est exécuté. Le code utilise la bibliothèque PHPExcel pour créer un fichier Excel et y ajouter les données du tableau $tableData. Le fichier Excel est ensuite téléchargé par l'utilisateur.

 Script JavaScript : Le code JavaScript est vide dans cet exemple, mais il peut être utilisé pour ajouter des fonctionnalités supplémentaires à la page.

En résumé, cette page permet à l'utilisateur d'exporter les informations d'une instance vers un fichier Excel, facilitant ainsi la gestion et l'analyse des données dans un format tabulaire.



Export\_instance.tpl.php : Ajout d’un tableau et d’un conteneur pour séparer les donnée affichés sur la page

CreationOrganisme.php / creationelus.php

Les champs reviendrons alors visibles :





J'ai ajouté la classe table table-bordered pour styliser le tableau avec des bordures et la classe table-responsive pour permettre un défilement horizontal lorsque le tableau dépasse la largeur de l'écran.

J'ai également ajouté la classe btn btn-primary pour styliser le bouton d'exportation vers Excel.

# **III – Explications de la démarche réalisée**

Pour explication, les fichier contenu dans /template permettent la prise en charge d’afficher le contenu de la page (texte, colonnes,etc…) alors que les fichiers .Act sont des variable paramètres qui permettent l’exécution d’une action qui permet alors de rechercher les données et paramètres a prendre en comptes avant d’afficher la page web.

Généralement, dans une application web, les dépendances de fichiers peuvent être utilisées pour inclure des parties réutilisables de code, tels que des modèles de page ou des actions spécifiques. Voici une explication générale de la méthode de fonctionnement de ces dépendances :

* Dossier "/TEMPLATE" : Ce dossier peut contenir un modèle de page ou de mise en forme utilisé pour afficher le contenu d'une page spécifique. Il peut inclure des éléments HTML, des feuilles de style CSS, des scripts JavaScript, etc. Lorsqu'une page est générée, le code peut inclure le contenu de ce fichier "TEMPLATE" pour fournir une structure de base cohérente à la page.



* Dossier "/actions" : Ce dossier peut contenir des fichiers qui définissent des actions ou des fonctionnalités spécifiques de l'application. Ces fichiers peuvent être appelés depuis d'autres parties de l'application pour effectuer des opérations spécifiques, telles que le traitement de formulaires, l'accès à une base de données, la manipulation de données, etc. Les fichiers dans ce dossier peuvent être organisés selon une structure logique, telle que des fichiers regroupant des actions liées à un module particulier de l'application.



En utilisant ces dépendances de fichiers, l'application peut bénéficier de la modularité et de la réutilisabilité du code. Les fichiers "TEMPLATE" permettent de séparer la structure et le contenu de la page, facilitant ainsi la maintenance et les mises à jour. Les fichiers dans le dossier "/actions" permettent d'organiser les fonctionnalités de l'application en modules ou en actions spécifiques, ce qui rend le code plus lisible et plus facile à maintenir.

J’utilise également une bibliothèque interne PHPExcel 1.8.1 au serveur que j’ai commit via GitHub pour la conversion de tableau ou d’instance via un fichier Excel depuis la page externe a Administration. Donc uniquement accessible via les droits requis.



# IV – Veille technologique, idées d’amélioration

Changer l’interface du tableau pour le rendre plus lisible et proposer une modification directe des informations contenue (en php ajouter la modification d’un champ, ou la suppression depuis la base de données. Uniquement en droit administrateur)

Optimisé les requêtes SQL utilisées pour extraire les données de la base de données. Utiliser des index sur les colonnes fréquemment utilisées dans les requêtes, il faudra éviter les requêtes redondantes et utilisez des clauses WHERE appropriées pour limiter les résultats aux données nécessaires.

Offrir aux utilisateurs différentes options d'export pour répondre à leurs besoins. Par exemple, on peut permettre aux utilisateurs de sélectionner des filtres ou des critères spécifiques pour exporter uniquement les représentants qui les intéressent. Il est possible également d’inclure des options pour spécifier le format de fichier souhaité (par exemple, Excel ou CSV).

Gestion des erreurs et des exceptions : Veiller à gérer les erreurs et les exceptions de manière appropriée. Afficher des messages d'erreur clairs et informatifs en cas de problème (via une requête php) lors de l'exportation ou de la manipulation des données. Logger les erreurs pour faciliter le débogage ultérieur et s’assurer de mettre en place des mécanismes de récupération pour éviter les interruptions de service.

# V – Conclusion

A finir pour le 19/06/23

Export\_instance.tpl.php

* Affichage de toutes les informations d’instance (manque ID\_elus + décalage au niveau de la requête SQL qui ne s’exécute pas en totalité)