

Sous-direction de l'expertise

Bureau de la donnée

35 rue de la Gare

75019 PARIS

Plan d'amélioration de l'application **AGIR**
(**A**ide à la **G**estion des **I**ndicateurs à **R**isque)

Mars 2021

SOMMAIRE

INTRODUCTION	3
PROJET.....	4
RESPECT DU CCLG_CCT	4
SEPARATION DE L'IHM ET DE L'API.....	4
CODE.....	6
COMMENTAIRES DANS LE CODE	6
FACTORISATION DU CODE	6
ERGONOMIE.....	7
RENDRE L'APPLICATION WEB RESPONSIVE	7
AIDE A LA SAISIE	7
BASE DE DONNEES	8
COHERENCE ENTRE LA BASE DE DONNEES ET L'INTERFACE.....	8
STRUCTURE D'ACCES AUX DONNEES	8

Introduction

Ce document présente les modifications qui permettraient d'améliorer le fonctionnement et les performances de l'application AGIR, de faciliter son utilisation et de simplifier sa maintenance.

Le fonctionnement actuel d'AGIR répondant aux attentes en terme de performance et d'utilisation, la mise en œuvre de ces modifications n'est pas requise dans l'immédiat. Cependant, le projet souhaite communiquer sur « l'état » actuel de l'application et sur les solutions d'amélioration possibles. Celles-ci ont été identifiées par le prestataire sortant du marché.

Projet

Respect du CCLG_CCT

Dans le cadre de la mise aux normes de l'application AGIR (réalisée), certaines actions préconisées dans le CCLG_CCT ont finalement été écartées - en accord avec le bureau de l'architecture- en raison du contexte et des besoins du projet.

Mise aux normes du projet dans le respect du CCLG_CCT
Solution(s) proposée(s)
Tests de montée en charge (type JMeter) Qualimétrie (type SonarQube) Installation auto portante (type Maven) : un script d'installation existe Tests unitaire automatisés (type PHPUnit) Support RGAA
Avantage(s)
Respect du CCLG_CCT
Inconvénient(s) / Risque(s)
-
Prérequis
-

Séparation de l'IHM et de l'API

Séparation de l'IHM et de l'API
Solution(s) proposée(s)
Mettre en place le Framework Symfony côté API et AngularJS côté FRONT afin de séparer les fonctionnalités du serveur et du navigateur Ne plus enregistrer les entités HTML dans la base de données Industrialiser le site (« minifier » les CSS et JS, fusionner les JS, ...)
Avantage(s)
Améliorer les performances de l'application Possibilité de mettre en place des templates HTML réutilisables Faciliter la maintenance et les tests Sécuriser l'accès aux données
Inconvénient(s) / Risque(s)
Nécessite une réécriture de l'application car actuellement le code HTML est intégralement inclus dans le code PHP
Prérequis
-

Remarque : la réécriture de l'application n'est pas envisagée en raison du coût trop important que cela impliquerait.

Utilisation du mode SOAP/REST de Symfony
Solution(s) proposée(s)
Utiliser les fonctions SOAP/REST du Framework de Symfony
Avantage(s)
Sécuriser et optimiser le traitement en utilisant des standards du web (XML pour la structure et HTTP pour le transport) pour les échanges avec la plateforme d'échange de fichiers (PFE)

Inconvénient(s) / Risque(s)
-
Prérequis
Séparation de l'IHM et de l'API pour permettre l'utilisation des fonctionnalités de Symfony

Code

Commentaires dans le code

Améliorer les commentaires dans le code
Solution(s) proposée(s)
Ajouter des commentaires sur l'ensemble des fonctions complexes du code
Avantage(s)
Faciliter la maintenance de l'application et gagner du temps lors des développements en évitant le recours à la consultation de la documentation technique
Inconvénient(s) / Risque(s)
Nécessite une bonne connaissance du projet et du code existant
Prérequis
-

Remarque : des commentaires ont déjà été ajoutés sur les parties de code impactées par des évolutions

Factorisation du code

Factorisation du code
Solution(s) proposée(s)
Supprimer les parties de code en doublon et créer des fonctions globales
Avantage(s)
Simplifier le code et limiter sa duplication
Inconvénient(s) / Risque(s)
Nécessite une bonne connaissance du projet et du code existant
Prérequis
-

Remarque : la factorisation du code a déjà été réalisée sur les parties impactées par des évolutions

Rendre l'application web responsive

Adaptation du contenu des pages à la taille de l'écran
Solution(s) proposée(s)
Mettre en place la bibliothèque Bootstrap dans l'application
Avantage(s)
Meilleure ergonomie générale de l'application Adapter automatiquement le contenu des pages à la taille de l'écran Agrandissement/rétrécissement du navigateur sans rechargement graphique (concerne l'écran de saisie) Portabilité de l'application sur l'ensemble des navigateurs
Inconvénient(s) / Risque(s)
Nécessite la modification de l'ensemble du code HTML et des styles du projet Besoins de tests approfondis sur l'affichage avec différentes tailles d'écran après la mise en œuvre
Prérequis
Séparation de l'IHM et de l'API

Aide à la saisie

Amélioration de l'aide à la saisie
Solution(s) proposée(s)
Mettre en place un typage et un masque de saisie sur les champs qui en sont dépourvus Mettre en place l'auto-complétion lorsque cela est possible (ex : calendrier) Développer l'utilisation de l'AJAX (peu d'AJAX actuellement dans le projet)
Avantage(s)
Eviter les erreurs de saisies par les utilisateurs Diminuer le nombre d'appels au serveur (les données sont vérifiées lors de la validation des formulaires)
Inconvénient(s) / Risque(s)
-
Prérequis
-

Remarque : l'aide à la saisie a été mise en place partiellement

Base de données

Cohérence entre la base de données et l'interface

Suppression des problèmes sémantiques sur les termes périodes et campagne
Solution(s) proposée(s)
Inverser le nom des tables « périodecollecte » et « campagnecollecte » (les campagnes sont enregistrées dans la table « périodecollecte » et les périodes dans la table « campagnecollecte ») Revoir la structure de la base de données (recours actuel à un paramétrage et une procédure de calcul complexe)
Avantage(s)
Faciliter la compréhension de la base de données par les développeurs Limiter le risque d'erreur lié à l'inversion « période » / « campagne » Faciliter l'évolutivité de l'application Optimiser l'application
Inconvénient(s) / Risque(s)
Nécessité d'utiliser des tables intermédiaires et de vérifier l'ensemble des contraintes d'intégrité
Prérequis
-

Structure d'accès aux données

Revoir la structure d'accès aux données
Solution(s) proposée(s)
Limiter le nombre de requêtes et mieux les filtrer dans les traitements
Avantage(s)
Diminuer du nombre de flux entre l'application et la base de données : amélioration des performances de l'application / diminution des temps de réponses Faciliter la maintenance
Inconvénient(s) / Risque(s)
Risque de régression
Prérequis
-

Remarque : ces modifications ont été réalisées partiellement