

Documentation technique

Gestion des reports

Contenu

[Vues](#)

[Contrôleurs](#)

Version : 1.0.0.2

Vues :

Nom	Descriptif
vHeader	affichage de l'en-tête de la page
vListeMessage	liste des messages liés à un report
vRechercheReport	formulaire de recherche d'un report
vSaisieReport	formulaire de saisie d'un nouveau report
vAddMessage	formulaire de saisie d'un nouveau message

Contrôleurs :**1. cReport/indexSaisieReport**

Fonctionnalité	Descriptif
Gestion des saisies de reports	Contrôleur de gestion des saisies de reports
	Affichage de la vue vHeader
	Appel de la méthode saisieReport
	Affichage de la vue vFooter

2. cReport/indexListeReport

Fonctionnalité	Descriptif
Gestion du listage des reports	Contrôleur de gestion du listage des reports
	Si l'utilisateur est connecté :
	<ul style="list-style-type: none"> chargement de la vue vHeader
	Si l'utilisateur n'est pas connecté :
	<ul style="list-style-type: none"> appel de la méthode headerNonConnecte du contrôleur cConnexion
	Appel de la méthode listeReport

3. cReport/indexMesReports

Fonctionnalité	Descriptif
Gestion du listage des reports de l'utilisateur connecté	Contrôleur de gestion du listage des reports de l'utilisateur connecté
	Affichage de la vue vHeader
	Appel de la méthode mesReports

Version : 1.0.0.2

4. cReport/indexMesAlertes

Fonctionnalité	Descriptif
Gestion du listage des alertes du développeur connecté	Contrôleur de gestion du listage des alertes du développeur connecté
	Appel de la méthode <code>mesAlertes</code>

5. cReport/saisieReport

Fonctionnalité	Descriptif
Saisie d'un report	Affichage du formulaire pour la saisie d'un report
	Récupération de l'id de l'éventuelle application sélectionnée par l'utilisateur : <code>\$idSelectedApp</code>
	Chargement de toutes les applications <code>\$apps</code>
	Récupération des applications qui ont au moins 1 cas d'utilisation et stockage dans <code>\$appsAAfficher</code>
	Mise en place logique client <code>.application -> saisieReport</code> <code>.application -> choixUseCaseSaisieReport</code> <code>.application -> hide</code>
	Passage des variables et affichage de la vue <code>vSaisieReport</code>

6. cReport/listeReport

Fonctionnalité	Descriptif
Listage des reports	Affichage du formulaire pour le listage des reports
	Récupération de l'id de l'éventuelle application sélectionnée par l'utilisateur : <code>\$idSelectedApp</code>
	Chargement de toutes les applications <code>\$apps</code>
	Récupération des applications qui ont au moins 1 cas d'utilisation et stockage dans <code>\$appsAAfficher</code>
	Mise en place logique client <code>.application -> listeReport</code> <code>.application -> choixUseCaseListeReport</code>
	Passage des variables et affichage de la vue <code>vRechercheReport</code>

Version : 1.0.0.2

7. cReport/choixUseCase

Fonctionnalité	Descriptif
Choix d'un cas d'utilisation	Méthode utilisée pour l'affichage des cas d'utilisation d'une application sélectionnée
	Récupération de l'id du use case sélectionné : <code>\$idUsecase</code>
	Chargement des cas d'utilisation de l'application : <code>\$useCases</code>
	S'il y a des cas d'utilisation : -> appel de la fonction <code>affichageRadioButton</code> de la classe technique <code>Gui</code> Sinon : -> affichage message indiquant qu'il n'y a pas de cas d'utilisation existant

8. cReport/choixUseCaseSaisieReport

Fonctionnalité	Descriptif
Choix d'un cas d'utilisation	Méthode utilisée pour l'affichage des cas d'utilisation d'une application sélectionnée, dans le cas de la saisie d'un report
	Récupération de l'id de l'application sélectionnée : <code>\$idApplication</code>
	Appel de la méthode <code>choixUseCase</code>
	Mise en place logique client <code>.useCase</code> -> <code>choixUseCase</code> <code>.useCase</code> -> <code>afficherComposants</code>

9. cReport/choixUseCaseListeReport

Fonctionnalité	Descriptif
Choix d'un cas d'utilisation	Méthode utilisée pour l'affichage des cas d'utilisation d'une application sélectionnée, dans le cas du listage des reports
	Récupération de l'id de l'application sélectionnée : <code>\$idApplication</code>
	Appel de la méthode <code>choixUseCase</code>
	Mise en place logique client <code>.useCase</code> -> <code>choixUseCase</code> <code>.useCase</code> -> <code>afficherValidation</code>

Version : 1.0.0.2

10. cReport/afficherComposants

Fonctionnalité	Descriptif
Affichage des composants de formulaire	Affichage des composants de formulaire dans le cas de la saisie d'un report
	Appel de la méthode <code>affichageComposantTexte</code> de la classe technique <code>Gui</code> pour la zone de saisie du libellé
	Appels de la méthode <code>affichageComposantTexteArea</code> de la classe technique <code>Gui</code> pour les zones de saisie du descriptif, de l'action utilisateur, et du message système
	Envoi en champ caché de la date du jour
	Affichage des boutons de validation et d'annulation de la saisie
	Mise en place logique client <code>#descriptif</code> -> <code>afficheReportsSimilaires</code> <code>#userAction</code> -> <code>afficheReportsSimilaires</code> <code>#systemMessage</code> -> <code>afficheReportsSimilaires</code> <code>#btValidAddReport</code> -> <code>add</code> <code>#btCancelAddReport</code> -> <code>saisieReport</code>

11. cReport/afficherValidation

Fonctionnalité	Descriptif
Affichage des composants de formulaire	Affichage du bouton de validation dans le cas du listage des reports
	Mise en place logique client <code>#btValidRechercheReport</code> -> <code>recherche</code>

12. cReport/afficheReportsSimilaires

Fonctionnalité	Descriptif
Affichage reports similaires	Affichage des reports similaires lors de la saisie d'un report
	Instanciation d'un nouveau report : <code>\$report</code>
	Affectation des données saisies aux membres de <code>\$report</code>
	Chargement de tous les reports existant : <code>\$reports</code>
	Instanciation d'un nouveau moteur de recherche : <code>\$searchEngine</code>
	Parcours de tous les reports et utilisation de la méthode <code>compare()</code> de la classe <code>SearchEngine</code> pour déterminer les reports similaires
Affichages des reports similaires, s'il y en a	

Version : 1.0.0.2

13. cReport/add

Fonctionnalité	Descriptif
Ajout	Ajout d'un report dans la base d edonnées
	Instanciation d'un nouveau report : \$report
	Affectation des données saisies aux membres de \$report
	Instanciation d'un objet Utilisateur, Statut, et UseCase affectés aux membres correspondants de \$report
	Vérification que tous les champs du report sont renseignés Si un des champs n'est pas rempli : <ul style="list-style-type: none"> affichage d'un message d'erreur Sinon : <ul style="list-style-type: none"> insertion dans la base de données affichage du message d'insertion

14. cReport/recherche

Fonctionnalité	Descriptif
Affichage reports	Affichage des reports trouvés en fonction de l'application et du cas d'utilisation choisis
	Récupération de l'id du cas d'utilisation choisi : \$idUseCase
	Chargement de tous les reports concernant le use case
	S'il y des reports trouvés : <ul style="list-style-type: none"> ->affichages des reports en appelant la méthode selectReport de la classe technique Gui Sinon : <ul style="list-style-type: none"> ->affichage d'un message indiquant qu'aucun report n'a été trouvé

15. cReport/afficherReport

Fonctionnalité	Descriptif
Affichage report sélectionné	Affichage du report sélectionné
	Récupération de l'id du report sélectionné : \$idReport
	Chargement du report : \$report
	Mise en place logique client #btValidChoiceMessage -> createMessage
	Passage des valeurs et chargement de la vue vListeMessage

16. cReport/createMessage

Fonctionnalité	Descriptif
Création d'un message	Affichage du formulaire d'envoi d'un message
	Mise en place logique client #btValidAddMessage -> addMessage
	Passage des valeurs et chargement de la vue vAddMessage

Version : 1.0.0.2

17. cReport/addMessage

Fonctionnalité	Descriptif
Ajout	Ajout d'un message pour un report
	Vérification que tous les champs sont remplis
	Si tous les champs ne sont pas remplis : <ul style="list-style-type: none"> • affichage d'un message d'erreur
	Si tous les champs sont remplis : <ul style="list-style-type: none"> -> si c'est une réponse officielle : <ul style="list-style-type: none"> • récupération du report concerné : \$report • affectation de la réponse officielle au report • changement du statut du report • mise à jour du report dans la base de données • affichage d'un message de mise à jour -> si c'est un message "simple" : <ul style="list-style-type: none"> • instanciation d'un objet Message : \$message • affectation des données du POST aux membres du message • affectation de la date d'envoi à \$message • instanciation d'un objet Report et Utilisateur et affectation à \$message • affichage du message d'insertion

18. cReport/mesReports

Fonctionnalité	Descriptif
Reports de l'utilisateur connecté	Récupération de l'ensemble des reports de l'utilisateur connecté à l'application
	Récupération des reports et stockage dans une variable : \$reports
	Mise en place logique client .reportSelected -> afficherReport
	Chargement de la vue vMesReports et passage des paramètres

19. cReport/mesAlertes

Fonctionnalité	Descriptif
Alertes du développeur connecté	Récupération de l'ensemble des alertes du développeur connecté à l'application
	Récupération de tous les reports non résolus et stockage dans une variable : \$lesReports
	Récupération des applications sur lesquelles le développeur est affecté : \$lesApplicationsConcernees
	Récupération des reports qui concernent ces applications : \$lesReportsConcernes
	Mise en place logique client .reportSelected -> afficherReport
	Chargement de la vue vMesAlertes et passage des paramètres