

## Documentation technique

---

### Connexion à l'application

## Contenu

---

[Variables de session](#)

[Vues](#)

[Contrôleurs](#)

Version : 1.0.0.2

**Variables de session (\$ SESSION) :**

Nom	Type	Descriptif
<code>\$_SESSION["userConnecte"]</code>	Instance de la classe utilisateur	Contient l'instance de la classe utilisateur qui correspond à l'utilisateur connecté à l'application
<code>\$_SESSION["lesPages"]</code>	Tableau d'instances de pages	Contient l'ensemble des pages auxquelles l'utilisateur connecté peut accéder
<code>\$_SESSION["lesPagesPublic"]</code>	Tableau d'instances de pages	Contient l'ensemble des pages auxquelles un utilisateur non connecté peut accéder

**Vues :**

Nom	Descriptif
vConnexion	affichage de la page de connexion à l'application
vAccueil	affichage de la page d'accueil sur l'application

**Contrôleurs :****1. cConnexion/index**

Fonctionnalité	Descriptif
<b>Lancement de l'application</b>	<b>Contrôleur du lancement de l'application</b>
	Destruction de la session pour effacer toutes les données qui y seraient enregistrées
	Ouverture du fichier de configuration contenant le nom de la base de données
	Lecture du nom de la base de données contenu dans le fichier de configuration Si la base existe sur le serveur : <ul style="list-style-type: none"> <li>• récupération de toutes les pages accessibles à un utilisateur non connecté et stockage dans la variable de session : <code>\$_SESSION["lesPagesPublic"]</code></li> <li>• appel de la méthode <code>indexListeReport</code> du contrôleur <code>cReport</code></li> </ul> Si la base n'existe pas sur le serveur : <ul style="list-style-type: none"> <li>• appel de la vue <code>vInstaller</code></li> </ul>

Version : 1.0.0.2

## 2. cConnexion/authentification

Fonctionnalité	Descriptif
Connexion à l'application	<b>Contrôleur de gestion des connexions</b>
	Destruction de l'éventuel contenu de la variable de session \$_SESSION['userConnecte']
	Mise en place logique client #btValidConnect -> verifConnexion #btCancelConnect -> authentication #createAccount -> createAccount
	Passage de variables (\$data[], \$erreur) et affichage de la vue vAuthentification

## 3. cConnexion/createAccount

Fonctionnalité	Descriptif
Création d'un compte	<b>Création d'un compte utilisateur pour se connecter à l'application</b>
	Appel de la méthode headerNonConnecte
	Mise en place logique client #btValidAddUser -> addAccount #btCancelAddUser -> createAccount
	Passage de variables (\$js) et affichage de la vue vCreateAccount

## 4. cConnexion/addAccount

Fonctionnalité	Descriptif
Ajout d'un compte	<b>Ajout d'un compte utilisateur</b>
	Instanciation d'un nouvel utilisateur et affectation des données du POST à ses membres
	Insertion du nouvel utilisateur dans la base de données

## 5. cConnexion/headerNonConnecte

Fonctionnalité	Descriptif
Header utilisateur non connecté	<b>Affichage des pages auxquelles un utilisateur non connecté à accès</b>
	Récupération des pages accessibles à un utilisateur non connecté
	Passage de variables (\$data[]) et affichage de la vue vConnexion

Version : 1.0.0.2

**6. cConnexion/verifConnexion**

Fonctionnalité	Descriptif
Vérification des identifiants saisis	<b>Utilisé pour vérifier la cohérence des identifiants de connexion saisis</b>
	Création d'une instance de classe utilisateur, <code>\$user</code> , avec les identifiants saisis
	Vérification de l'existence de l'utilisateur dans la base de données
	Si l'utilisateur existe : <ul style="list-style-type: none"> <li>-&gt; si l'utilisateur appartient à un groupe qui dispose de droits : <ul style="list-style-type: none"> <li>• on stocke l'instance de la classe utilisateur dans une variable de session : <code>\$_SESSION['userConnecte']</code></li> <li>• on récupère son groupe</li> <li>• on récupère toutes les pages auxquelles il a accès, et on les stocke dans une variable de session : <code>\$_SESSION['lesPages']</code></li> <li>• appel de la méthode <code>accueil</code></li> </ul> </li> <li>-&gt; si l'utilisateur appartient à un groupe qui ne dispose pas de droits : <ul style="list-style-type: none"> <li>• affichage d'un message d'erreur</li> </ul> </li> </ul>
Si l'utilisateur n'existe pas : <ul style="list-style-type: none"> <li>• on appelle la méthode <code>index</code>, en lui passant la variable <code>\$erreur</code> initialisée à "Erreur de connexion"</li> </ul>	

**7. cConnexion/accueil :**

Fonctionnalité	Descriptif
Accueil	<b>Utilisé pour charger la page d'accueil</b>
	appel de la méthode <code>indexMesReports</code> du contrôleur <code>cReport</code> si l'utilisateur est un développeur : appel de la méthode <code>indexMesAlertes</code> du contrôleur <code>cReport</code>