

Installation WebRSA 4

Client	CD
Projet	WebRSA
Auteur	E. DUCHESNE / E. PRITCHARD / M. MASY P. LAVIGNE Atol C&D
Objet de l'installation	Configuration WebRSA 4
Date de l'intervention	
Version(s) installée(s)	Version 4
Date de révision du document	16 juin 2020
ID machine	
Contenu	Voir détail ci-après

Table des matières

1. Application	3
1.1 Mise en place de la base de données	3
1.2 Application	3
1.3 Ajustement des droits de l'utilisateur	4
1.4 Variables d'environnement	4
1.5 Variables de configuration	5
1.6 Modèles des fichiers odt	5
2. Shell	5
3. Check de l'application	5
4. Script de pré chargement	6
5. Connexion	6

1. Application

1.1 Mise en place de la base de données

Pour le démarrage de l'application, il faut mettre en place la structure des données et les données minimales servant à la connexion d'un utilisateur.

Nous partons ici du postulat que nous sommes à la racine du projet WebRSA (*/{{Répertoire/d/installation}}/public_html*) :

Voici les commandes pour la mise en place de la structure de la base de données :

```
sudo su postgres
psql -U webrsa -d webrsa_web1 -h 127.0.0.1 -f project/app/Config/sql/webrsa.SCHEMA.public.sql
psql -U webrsa -d webrsa_web1 -h 127.0.0.1 -f
project/app/Config/sql/webrsa.SCHEMA.administration.sql
```

Voici la commande pour la mise en place des données minimales de la base de données :

```
psql -U webrsa -d webrsa_web1 -h 127.0.0.1 -f project/app/Config/sql/webrsa.DATA.public.CG99X.sql
```

1.2 Application

L'application est stockée dans :

```
/{{Répertoire/d/installation}}/public_html/
```

Pour installer l'application, vous devez suivre les instructions suivantes :

```
cd /var/www/webrsa-web1/
rm public_html/* -rf
tar zxvf webrsa.tar.gz -C public_html/

cd /var/www/webrsa-web1/public_html/
composer install
sudo app/webrsa.sh deploy
```

1.3 Ajustement des droits de l'utilisateur

La commande précédente (*sudo app/webrsa.sh deploy*) permet de calculer les droits minimaux des

utilisateurs, appeler ACO / ARO. Pour plus d'informations, vous pouvez consulter cette page <https://book.cakephp.org/2/en/core-libraries/components/access-control-lists.html>.

Comme il n'y a qu'un utilisateur qui ne s'est jamais connecté, il faut ajuster ces droits pour avoir accès à toute l'application et faire la liaison des ARO et ACO.

```
sudo su postgres
psql webrsa_web1
-- Récupérer l'ID de la table public.aros
SELECT id FROM public.aros WHERE model ILIKE 'group';
-- Récupérer l'ID de la table public.acos
SELECT id FROM public.acos WHERE alias ILIKE 'groups';
-- Faire la liaison entre les deux et donner les droits {ID_ARO} et {ID_ACO} sont à
-- remplacer avec les valeurs précédemment trouvées
INSERT INTO public.aros_acos(aro_id, aco_id, "_create", "_delete", "_read", "_update" )
VALUES({ID_ARO},{ID_ACO}, 1, 1, 1, 1);
```

Vous devez ensuite recharger les droits utilisateurs avec la commande suivante :

```
sudo app/webrsa.sh acoaro
```

1.4 Variables d'environnement

Les variables d'environnement sont dans ce fichier :

```
{Répertoire/d/installation}/.env
```

La liste des variables d'environnement se trouve dans le fichier **app/Config/.env.default**.

On peut repérer plusieurs groupes :

- Les variables **PRODUCTION** et **CD_DEPARTEMENT** qui permettent de définir si l'application est mode production (valeur = 1) ou non (valeur = 0) et le numéro du département.
- Les variables commençant par **CMIS_** qui permettent de configurer la connexion avec un système de gestion de contenu comme Alfresco.
- Les variables commençant par **GEDOOO_** qui permettent de configurer la connexion avec Gedooo.
- Les variables commençant par **DATABASE_CONFIG_** qui permettent de configurer les variables de connexion aux bases de données.

Vous pouvez remplir les valeurs en vous référant au fichier **app/Config/database.php**. Ce fichier **.env.default** répertorie toutes les connexions aux bases de données, il est donc probable que vous laissiez des valeurs vides.

- Les variables commençant par **EMAIL_CONFIG_** qui permettent de configurer les variables de configuration d'envoi de mail.

Vous pouvez remplir les valeurs en vous référant au fichier **app/Config/email.php**. Ce fichier **.env.default** répertorie toutes les configurations d'envoi de mail, il est donc probable que vous laissiez des valeurs vides.

1.5 Variables de configuration

Les variables de configuration sont définies :

- **Dans le fichier app/Config/webrsa.inc**
Ce fichier sert à injecter dans WebRSA les variables générales de base provenant du fichier `.env`. Les autres variables du fichier `.env` sont injectées dans les fichiers `app/Config/email.php` et `app/Config/database.php`.
- **En base de données**
Elle sont accessibles depuis le menu **Administration > Configuration** et regroupent toutes les variables de configuration de l'application.

1.6 Modèles des fichiers odt

Les modèles des fichiers odt sont ici :

```
{Répertoire/d/installation}/contrib/modelesodt/
```

Vous devez copier vos modèles odt dans ce répertoire.

2. Shell

Pour le bon fonctionnement des shells, vous devrez ajouter systématiquement `-app app` au shells que vous souhaitez exécuter si il ne sont pas déjà en fin d'instruction :

```
sudo -u apache vendor/cakephp/cakephp/lib/Cake/Console/cake -app app
```

3. Check de l'application

Le check de l'application sert à vérifier le bon fonctionnement de l'application et permet de remonter les éventuelles erreurs.

Pour lancer le check de l'application, vous devez exécuter cette commande :

```
sudo app/webrsa.sh check
```

Si une erreur de Posix survient, vous pouvez au choix :

- Installer le paquet PHP-Posix

- Remplacer 3 des lignes du fichier app/Plugin/Appchecks/Model/Check.php par les paramètres de votre système.

```
-- Lignes à remplacer
$user = posix_getpwuid(posix_geteuid());
$group = posix_getgrgid($user['gid']);
$username = posix_uname();

-- Exemple de nouvelles lignes :
$user = 'apache' ;
$group = 'apache' ;
$username = 'Linux' ;
```

Vous pouvez également faire cette vérification depuis l'application web depuis la page "Vérification de l'application" accessible à partir du menu "Administration".

4. Script de pré chargement

Le script de pré chargement de l'application sert à la prise en compte des modifications de la structure de la base de données par l'application.

Pour lancer le script de pré chargement de l'application, vous devez exécuter cette commande :

```
sudo app/webrsa.sh prechargement
```

5. Connexion

La première connexion se fait avec l'utilisateur prédéfini en base de données :

```
Login : webrsa
Mot de passe : webrsa#2020
```

Il est conseillé de changer ce mot de passe lors de la première connexion depuis le menu Administration > Habilitations > Utilisateurs.