

Framework - Documentation

Vue d'ensemble

Application web PHP légère utilisant SQLite pour la gestion de l'authentification et des utilisateurs. Cette solution est containerisée avec Docker et utilise Nginx comme serveur web.

Accès à l'application

- **URL** : https://localhost
- **Protocole** : HTTPS (configuration via nginx.conf)

Architecture technique


Base de données

- **Type** : SQLite (base de données embarquée)
- **Fichier** : UserData.db (créé automatiquement)
- **Avantages** : Pas de serveur DB séparé, configuration simplifiée

Sécurité

L'application implémente plusieurs mesures de sécurité :

- **Hachage des mots de passe** : Utilisation de bcrypt pour un stockage sécurisé
- **Gestion des sessions** : Authentification via sessions PHP natives
- **Contrôle d'accès** : Middleware d'authentification (auth.php)

 **Important** : Changez immédiatement le mot de passe administrateur par défaut après la première installation.

Compte administrateur par défaut

Lors de la première installation, un compte admin est créé avec les identifiants suivants :

Champ	Valeur
Utilisateur	admin
Mot de passe admin	

Fonctionnalités principales

Gestion des utilisateurs

- Connexion et déconnexion

- Création de nouveaux utilisateurs
- Gestion des droits d'accès
- Promotion d'utilisateurs en administrateurs

Administration

- Tableau de bord administrateur
- Gestion complète des comptes utilisateurs
- Contrôle des permissions

Déploiement avec Docker

Démarrage de l'application

`docker compose up -d --build`

Options :

- `-d` : Mode détaché (exécution en arrière-plan)
- `--build` : Reconstruction des images si nécessaire

Arrêt de l'application

`docker compose down`

Pour arrêter et supprimer les volumes :

`docker compose down -v`

Vérification du statut

`docker compose ps`

Consultation des logs

`docker compose logs -f`

Structure des fichiers

.

- |— `index.php` # Point d'entrée principal de l'application
- |— `auth.php` # Middleware d'authentification et contrôle d'accès
- |— `UserData.db` # Base de données SQLite (générée automatiquement)
- |— `nginx.conf` # Configuration du serveur web Nginx
- |— `Dockerfile` # Instructions de construction du conteneur

└─ docker-compose.yml # Orchestration des services Docker

Description des fichiers clés

Fichier	Rôle
index.php	Page d'accueil et routeur principal
auth.php	Vérification des sessions et redirections
UserData.db	Stockage des utilisateurs et permissions
nginx.conf	Règles de routage et configuration HTTPS

Sécurisation des pages

Protection par authentification

Pour protéger une page et exiger une authentification, ajoutez cette ligne **au tout début** de votre fichier PHP (avant tout code HTML) :

```
<?php
require 'auth.php';
?>
```

Exemple d'utilisation

Page protégée (dashboard.php) :

```
<?php
require 'auth.php';
?>

<!DOCTYPE html>

<html>

<head>

  <title>Tableau de bord</title>

</head>

<body>

  <h1>Bienvenue sur votre tableau de bord</h1>

  <p>Cette page est protégée et nécessite une authentification.</p>

</body>
```

</html>

Page publique (about.php) :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>À propos</title>
```

```
</head>
```

```
<body>
```

```
  <h1>À propos de nous</h1>
```

```
  <p>Cette page est accessible sans connexion.</p>
```

```
</body>
```

```
</html>
```

Comportement du système d'authentification

Lorsque auth.php est inclus dans une page :

1. **Vérification de session** : Le script vérifie si l'utilisateur est connecté
2. **Redirection automatique** : Si non connecté, l'utilisateur est redirigé vers la page de connexion
3. **Accès autorisé** : Si connecté, la page s'affiche normalement

Bonnes pratiques

✓ À faire :

- Placer require 'auth.php'; en première ligne du fichier PHP
- Protéger toutes les pages contenant des données sensibles
- Protéger les pages d'administration et de gestion

✗ À éviter :

- Placer l'include après du code HTML
- Oublier de protéger les pages contenant des fonctionnalités critiques
- Utiliser include au lieu de require (require génère une erreur fatale si le fichier manque)

Pages typiquement à protéger

- ✓ dashboard.php # Tableau de bord utilisateur
- ✓ profile.php # Profil utilisateur
- ✓ admin/users.php # Gestion des utilisateurs
- ✓ admin/settings.php # Paramètres système
- ✓ api/data.php # Endpoints API sensibles
- X index.php # Page d'accueil publique
- X login.php # Page de connexion
- X about.php # Page à propos

Configuration et personnalisation

Modifier la configuration Nginx

Éditez le fichier nginx.conf pour :

- Changer le port d'écoute
- Configurer les certificats SSL
- Ajuster les règles de réécriture d'URL

Variables d'environnement

Les variables peuvent être définies dans le fichier docker-compose.yml :

environment:

- PHP_MEMORY_LIMIT=256M
- PHP_UPLOAD_MAX_FILESIZE=10M

Recommandations de sécurité

1. **Changez le mot de passe admin** dès la première connexion
2. **Configurez des certificats SSL valides** pour la production
3. **Limitez l'accès réseau** via les règles firewall
4. **Effectuez des sauvegardes régulières** de UserData.db
5. **Mettez à jour régulièrement** les dépendances et l'image Docker

Sauvegarde et restauration

Sauvegarde de la base de données

```
docker compose cp <container_name>:/path/to/UserData.db ./backup/
```

Restauration

```
docker compose cp ./backup/UserData.db <container_name>:/path/to/
```

Dépannage

L'application ne démarre pas

Vérifier les logs

```
docker compose logs
```

Reconstruire complètement

```
docker compose down
```

```
docker compose build --no-cache
```

```
docker compose up -d
```

Erreur de connexion à la base de données

Vérifiez les permissions du fichier UserData.db et assurez-vous que le répertoire est accessible en écriture par le conteneur.

Impossible d'accéder via HTTPS

Vérifiez que :

- Le port 443 est bien mappé dans docker-compose.yml
- Les certificats SSL sont correctement configurés dans nginx.conf
- Aucun pare-feu ne bloque le port

Support et contribution

Pour signaler un problème ou suggérer une amélioration, veuillez créer une issue dans le dépôt du projet.