
DOCUMENTATION - TECHNIQUE

Table des matières

▸ Informations sur le projet & CDC	Page N°1
▸ Identifiants de connexion	Page N°2
▸ Info & structure de la base de données	Page N°3
▸ Arborescence du projet	Page N°4
▸ Tableau des routes	Page N°5-6
▸ Système d'authentification	Page N°8
▸ Fonctions personnalisées	Page N°9
▸ Mettre en prod le projet	Page N°10
▸ Infos mails & docs	Page N°11
▸ Tâches CRON	Page N°12
▸ RGPD	Page N°13

→ Informations & CDC

AteRisk Web est une application web Symfony permettant la gestion de rendez-vous des patients de l'hôpital. Il contient un système de connexion pour les patients afin de gérer et de voir leurs rendez-vous puis un système de connexion pour les membres du personnel :

- Un assistant peut confirmer, modifier ou supprimer un rendez-vous.
- Un médecin peut afficher ses rendez-vous.

Un patient pourra donc choisir un rendez-vous avec un médecin selon ses disponibilités.

Système de mail permettant d'avertir le patient lorsque qu'un rdv est validé, modifié, annulé ou supprimé.

Un administrateur pourra gérer les différents comptes des personnels, il pourra donc en ajouter, modifier et supprimer.

→ Identifiants de tests

Patients

Username : *patient1@patient.fr, patient2@patient.fr*

Mot de passe : *patient1, patient2*

Infos : xx

Assistants

Username : *assistant@assistant.fr, martin.dupont@gmail.com, kelis.kevin@free.fr*

Mot de passe : *assistant2023*

Infos : Les users ayant pour rôle assistant on le même mot de passe

Médecins

Username : *medecin@medecin.fr, marie.dubois@gmail.com, propre.monsieur@chauve.fr*

Mot de passe : *medecin2023*

Infos : Les users ayant pour rôle médecin on le même mot de passe

Administrateurs

Username : *admin@admin.fr, rdlre.alex@gmail.com, patricia.jesaipo@orange.fr, julien.renard@gmail.com*

Mot de passe : *admin2023*

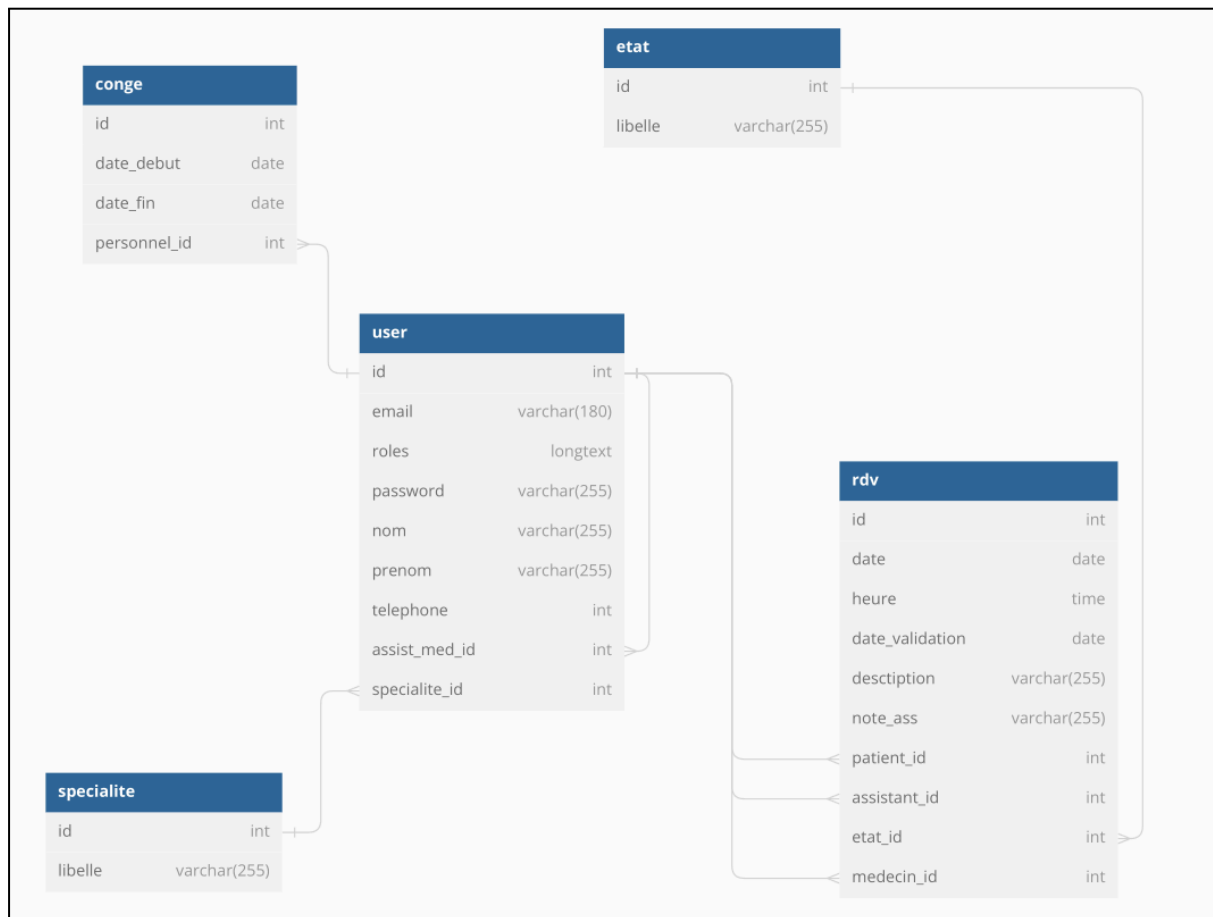
Infos : Les users ayant pour rôle admin on le même mot de passe

→ Infos & structure de la base de données

INFOS

- SGBD utilisé: MySQL (Hébergé en ligne sur Hostinger)
- IP: 154.49.245.52:3306
- User: u207216228_ateriskweb | Pass: **Admin2023***
- Database name: u207216228_ateriskweb
- Lien phpMyAdmin: <https://auth-db991.hstgr.io/>

STRUCTURES



→ Arborescence du projet

L'arborescence du projet est simple. Il est composé des dossiers et fichiers de bases de symfony puis de ceux créés pour le projet.

Dans le dossier public, se trouvent les dossiers :

- **css** -> contient les fichiers styles du site, triés par zone d'affichage :
 - **dashboard** contenant le css du panel de gestion côté personnel.
 - **patient** contenant le css de l'affichage côté patient.
- **images** -> contient toutes les images du site.
- **js** -> javascript contenant le javascript (trié comme pour le css).
- **scss** -> les fichiers scss utiles.
- **vendor** -> les templates nécessaires au bon fonctionnement du site.

Dans le dossier templates, se trouvent l'affichage :

- Dans le dossier **patient** : les twig côté patient.
- Dans le dossier **personnel** : plusieurs sous-dossiers contenant pour chacun d'eux, les twig reliés aux rôles des personnels.
Ex: le sous-dossier assistant contient toutes les vues côté assistant.

Dans le dossier src/Controller, se trouvent les contrôleurs. Chaque contrôleur est relié à un rôle utilisateur ou bien à un système défini. Par exemple, le contrôleur "AssistantController" contient tous les contrôles effectués reliés au rôle Assistant comme par exemple l'affichage des rendez-vous en attente de validation. (Voir tableau ci-dessous -> "Tableau des routes").

→ Tableau des routes

Contrôleur : MedecinController (Rôle accès : médecin)

Route	Nom associé	Page associée	Utilité
/medecin	medecin_index	personnel/medecin/index.html.twig	Page d'accueil médecin
/medecin/today	medecin_today	personnel/medecin/today.html.twig	Rdv du jour
/medecin/list	medecin_list	personnel/medecin/list.html.twig	Liste des rdv à venir
/medecin/historique	medecin_historique	personnel/medecin/historique.html.twig	Afficher l'historique de tous les rdv du médecin

Contrôleur : AssistantController (Rôle accès : assistant)

Route	Nom associé	Page associée	Utilité
/assistant	assistant_index	personnel/assistant/index.html.twig	Page d'accueil assistant
/assistant/waiting	assistant_waiting	personnel/assistant/waiting.html.twig	Rdv en attente de validation
/assistant/list	assistant_list	personnel/assistant/list.html.twig	Liste des rdv à venir
/assistant/historique	assistant_historique	personnel/assistant/historique.html.twig	Afficher l'historique de tous les rdv de son médecin

Contrôleur : PatientController (Rôle accès : tous)

Route	Nom associé	Page associée	Utilité
/patient	patient_index	patient/index.html.twig	Page d'accueil patient
/patient/rdv	patient_rdv	patient/rdv.html.twig	Prise de rendez-vous
/patient/rdv/{id}	patient_rdvshow	patient/rdvshow.html.twig	Afficher détails d'un rdv
/patient/list	patient_list	patient/list.html.twig	Liste des rdv à venir & en attente de valid
/patient/historique	patient_historique	patient/historique.html.twig	Afficher historique de tous les rdv du patient
/patient/account	patient_account	patient/account.html.twig	Page compte du patient
/patient/contact	patient_contact	patient/contact.html.twig	Page contact
/patient/rdv/prise/{id}	patient_priserdv	patient/priseRdv.html.twig	Prendre un rdv avec un médecin spécifique

Contrôleur : AdminController (Rôle accès : admin)

Route	Nom associé	Page associée	Utilité
/admin	admin_index	personnel/admin/index.html.twig	Page d'accueil admin
/admin/users/{role}	admin_updateUsers	personnel/admin/updateUsers.html.twig	Permet de gérer et d'afficher les comptes selon le rôle passé en paramètre

Contrôleur : RdvController (Rôle accès : assistant)

Route	Nom associé	Page associée	Utilité
/rdv/validate/{id}	rdv_validate	Aucune (redirige directement vers la page /assistant_waiting)	Mettre en état "Validé" le rdv en paramètre
/rdv/decline/{id}	rdv_decline	Aucune (redirige directement vers la page /assistant_waiting)	Mettre en état "Refusé" le rdv en paramètre
/rdv/cancel/{id}	rdv_cancel	Aucune (redirige directement vers la page /assistant_waiting)	Mettre en état "Annulé" le rdv en paramètre

Contrôleur : SecurityController (Rôle accès : tous)

Route	Nom associé	Page associée	Utilité
/login	app_login	security/login.html.twig	Page de connexion
/logout	app_logout	Aucune (redirige directement vers la page /login une fois déconnecté)	Page de déconnexion

Contrôleur : MainController (Rôle accès : tous)

Route	Nom associé	Page associée	Utilité
/	main	Aucune (redirige vers la page login)	Utile pour la mise en prod (permet de rediriger aterisk.eu vers aterisk.eu/login)

→ Ajout de fonctions personnalisées

Des fonctions personnalisées ont été ajoutées en plus de celles générées automatiquement par Doctrine.

- **getRole(\$user)**

- PARAM: L'utilisateur connecté
- DESC: *recupérer l'objet rôle de l'utilisateur*
- RETOUR: *l'objet rôle*
- LOC: UserRepository

- **getRoleToString(\$user)**

- PARAM: L'utilisateur connecté
- DESC: *recupérer le rôle de l'utilisateur et renvoyer à quoi il correspond (patient, médecin, assistant ou admin)*
- RETOUR: *le libelle du rôle*
- LOC: UserRepository

- **countUsersByRole(\$role)**

- PARAM: Un rôle (ex: ROLE_ADMIN)
- DESC: *recupérer le nombre d'utilisateur ayant le rôle en paramètre.*
- RETOUR: *le nombre d'utilisateur*
- LOC: UserRepository

- **findByRole(\$role)**

- PARAM: Un rôle (ex: ROLE_ADMIN)
- DESC: *recupérer tous les utilisateurs ayant le rôle en paramètre.*
- RETOUR: *les utilisateurs*
- LOC: UserRepository

→ Système d'authentification

Le processus d'authentification dans Symfony suit généralement les étapes suivantes :

- Requête d'authentification: L'utilisateur soumet ses informations d'identification, telles que le nom d'utilisateur et le mot de passe, via un formulaire ou tout autre moyen approprié.
- Authenticator: Un Guard Authenticator intercepte la requête d'authentification et détermine la manière dont l'authentification doit être gérée. Il peut valider les informations d'identification, rediriger l'utilisateur vers une page de connexion, ou même déclencher des actions spécifiques en fonction des besoins de l'application.
- Provider d'Authentification: Une fois que les informations d'identification ont été validées par l'authenticator, le provider d'authentification est sollicité pour récupérer les détails de l'utilisateur à partir de la base de données.
- Création d'une Session: Si l'utilisateur est authentifié avec succès, une session est créée et l'utilisateur est considéré comme connecté. Ses informations d'identification peuvent être stockées dans la session pour les requêtes futures.

→ Mettre en production l'app sur Hostinger (aterisk.eu)

Penser à passer la version de php en 8.1 ou + si pas déjà fait. (Only Alex)
Suivre les étapes [ici](#).

Compte FTP pour accéder aux fichiers de aterisk.eu :

- IP: [REDACTED] - Password: [REDACTED]
- Username: [REDACTED] - Port: 21

Accès SSH pour accéder au site aterisk.eu en mode console :

- IP: [REDACTED] - Password: [REDACTED]
- Username: [REDACTED] - Port: 65002

→ ssh -p [REDACTED] puis cd domains/aterisk.eu/public_html

Une fois les manips réalisées, dans :

.htaccess de la racine :

```
<IfModule mod_rewrite.c>
RewriteEngine On

RewriteCond %{HTTP_HOST} ^aterisk\.eu [NC]
RewriteRule ^$ /login [R=301,L]

RewriteCond %{REQUEST_URI} !^/\.\well-known/acme-challenge/
RewriteCond %{REQUEST_URI} !^/\.\well-known/pki-validation/[A-F0-9]{32}\.txt(?:\ Comodo\ DCV)?$
RewriteRule ^public/index\.php [L]

RewriteCond %{HTTPS} !=on
RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]

RewriteRule ^(.*)$ public/$1 [L]
</IfModule>

# DO NOT REMOVE THIS LINE AND THE LINES BELOW ERRORPAGEID:Q8jRKBSRHH
ErrorDocument 404 /404.html
# DO NOT REMOVE THIS LINE AND THE LINES ABOVE Q8jRKBSRHH:ERRORPAGEID
```

.htaccess du dossier public :

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^ index.php [QSA,L]
</IfModule>

# DO NOT REMOVE THIS LINE AND THE LINES BELOW ERRORPAGEID:Q8jRKBSRHH
ErrorDocument 404 /404.html
# DO NOT REMOVE THIS LINE AND THE LINES ABOVE Q8jRKBSRHH:ERRORPAGEID
```

SI PROB:

```
php bin/console debug:router
php bin/console cache:clear
php bin/console debug:router | grep app_login
```

php bin/console router:match /login

→ Infos mails & docs

- Adresse: noreply@aterisk.eu
- Mdp: !L7299yL}0VWI3UF%

- Adresse: contact@aterisk.eu
- Mdp: m@Wi]P.L60]6;l

[Lien pour se connecter](#) au webmail.

Liens des différents google doc du projet Web

- Partie développement
 - Doc Tâches & Time - [Lien](#)
 - Doc Tests - [Lien](#)
- Partie utilisateurs
 - Doc Patients - [Lien](#)
 - Doc Assistants - [Lien](#)
 - Doc Médecin - [Lien](#)
 - Doc Admin - [Lien](#)

→ Tâches CRON

Utile: Créer une commande : php bin/console make:command BackupDatabase

→ Backup DataBase

- Chemin de la commande : /src/Command/BackupDatabaseCommand.php
- La sauvegarde a lieu : toutes les 12H soit deux fois par jour.
(à 00 heure & à 12 heures) (à mettre en tâche CRON sur le serveur)
- Nom du fichier sauvegardé : (Date_Heure.sql)
- Pour tester la commande : php bin/console BackupDatabase
- Dossier de sauvegarde : /backup/bdd

Tâches CRON: crontab -e 0 0 * * * cd domains/aterisk.eu/public_html && php
bin/console BackupDatabase
crontab -e 0 12 * * * cd domains/aterisk.eu/public_html && php
bin/console BackupDatabase
(à mettre sur le serveur hébergeant l'application)

scp backup/bdd/21-12-2023_15-31-40.sql prof@192.168.1.21:backup/bdd/

→ RGPD

La page politique de confidentialité se trouve dans le dossier *templates/patient/* et est accessible par l'utilisateur via un lien cliquable dans le footer.

Le registre des traitements se trouve dans le dossier des documentations du projet.