

# Document technique de Kontrol

## Backend

### Spécification

- Node → [v22.5.1](#).
- API → [Express.js](#).
- Package Manager → [pnpm](#) ou [npm](#).
- Socket (Chat) → [Socket.io](#).
- Ports API → 3001.
- Ports Socket → 3002.

### Détails sur l'arborescence

**api/data** → Données des membres du domaine.

**api/data/{user}/user.data.json** → Ce JSON contient les informations du compte (mot de passe, niveau de permission, nom d'utilisateur et url de la photo de profil).

**api/data/{user}/website\_fav.data.json** → Ce JSON contient les sites favoris de l'utilisateur (nom du site, url et url de l'image de présentation).

**api/data/{user}/storage/** → Il n'y a pas de formatage particulier ici. C'est une zone libre d'espace pour l'utilisateur

**api/data/{user}/galery/{album}/** → Ici, c'est strict. Le répertoire "galery" d'un utilisateur contient uniquement des sous-dossiers (album) prenant uniquement des images / vidéos.

**api/data/{user}/calendar/** → Pas encore fait.

**api/data/{user}/notes/** → Pas encore fait.

---

**api/logs** → Logs de la console (pas encore fait).

---

**api/reports** → Message envoyé par les membres d'un domaine au support technique (nous).

---

**api/routes/{nom}Routes.js** → Routes que le client utilisera afin de contacter le serveur.

**api/routes/PandoraRoutes.js** → Routes concernant le stockage de fichier dans l'application "Stockage".

**api/routes/PhotomatonRoutes.js** → Routes concernant le stockage des images / vidéos dans l'application "Galerie".

**api/routes/ReportRoutes.js** → Routes concernant le stockage de report utilisateur. Ou contacter l'assistance du site.

**api/routes/UsersRoutes.js** → Routes concernant la gestion des utilisateurs.

**api/routes/WebfavRoutes.js** → Routes qui gère les sites favoris de l'utilisateur.

---

**api/service/{nom}.js** → Contient uniquement les classes qui vont servir de "support" aux routes. En effet, les routes vont faire appel aux fonctions des classes stockées dans api/service.

**api/service/Pandora.js** → Service concernant le stockage de fichier dans l'application "Stockage".

**api/service/Photomaton.js** → Service concernant le stockage des images / vidéos dans l'application "Galerie".

**api/service/Report.js** → Service concernant le stockage de report utilisateur. Ou contacter l'assistance du site.

**api/service/Users.js** → Service concernant la gestion des utilisateurs.

**api/service/Webfav.js** → Service qui gère les sites favoris de l'utilisateur.

**api/service/Hermes.js** → Service qui gère les interactions messagerie.

---

**api/temp** → Dossier que [multer](#) utilisé pour stocker temporairement des fichiers en attentes d'ordre.

## Commandes

<b>pnpm run start</b>	Démarrer le serveur
<b>pnpm run forever</b>	Démarrer le serveur "pour toujours"
<b>pnpm run stop</b>	Stopper le serveur démarrer "pour toujours"

## Frontend

### Spécification

- [React.js](#) & [Vite](#)
- Librairie de Style → [Joy UI](#).
- Librairie d'icone → [MUI Icon](#).
- Requete API → [Axios](#).
- Connexion Socket → [socket.io-client](#).
- Port frontend → 5173.

### Détails sur l'arborescence

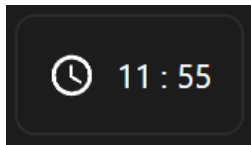
**public/** → Zone ou sur le site internet, on pourra accéder au fichier présent dans public via l'url. (exemple, si je met photo.jpg, elle sera accessible via <http://localhost:5173/photo.jpg>).

**src/assets/\*** → Contient les polices et les images non accessible via public.

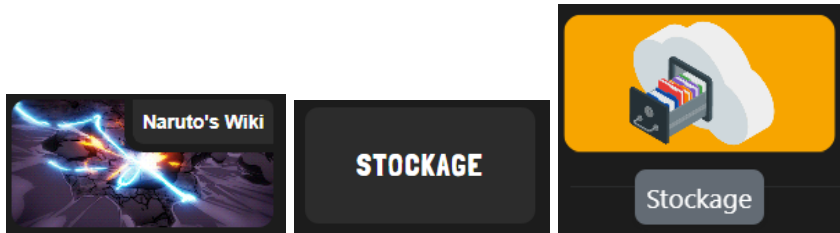
---

**src/components/{component}/{component}.jsx** → Les composants du site.

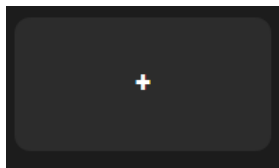
**src/components/Clock/Clock.jsx** → Horloge présent dans la navbar.



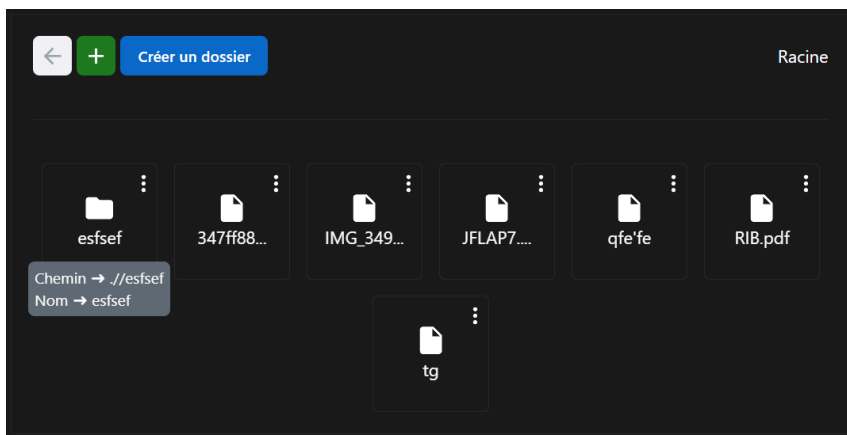
**src/components/Displayable/Displayable.jsx** → Rectangle utilisé un peu partout pour afficher un texte (et au survol un gif ou une image) ou une image avec un petit texte sur le coin supérieur droit.



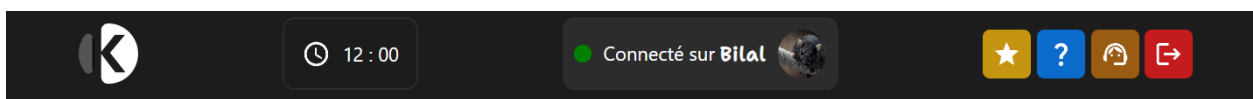
**src/components/DisplayableAdd/DisplayableAdd.jsx** → Un peu similaire au rectangle mais il y a un + a la place du texte.



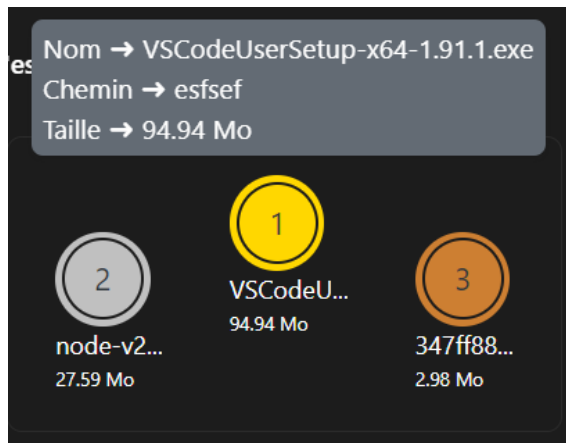
**src/components/FileManager/FileManager.jsx** → Composant qui sert d'explorateur de fichier.



**src/components/NavBar/NavBar.jsx** → Barre de navigation



**src/components/Olympe/Olympe.jsx** → Ce composant sélectionne les 3 plus gros fichiers du serveur pour en faire un classement.



**src/components/Pegaze/Pegaze.jsx** → Flèche présente partout sur le site permettant de revenir tout en haut.



**src/components/StorageDetailsBadge/StorageDetailsBadge.jsx** → Contient diverses informations sur le stockage.

Espace total	99999.00 Go
Espace libre	99998.87 Go
Espace utilisé	129.37 Mo

---

**src/GateContext/gate-context.jsx** → Contexte contenant les informations de base de l'utilisateur.

---

**src/hooks/useGate.js** → Hook (au même titre que useEffect par exemple) permettant de gérer le contexte gate-context.

---

**src/layout/MainLayout** → Layout majoritaire de l'application comprenant une navbar et un footer dans le futur.

---

**src/pages/{page}/{page}.jsx** → Pages de l'applications.

**src/pages/Chat/Chat.jsx** → Page de la messagerie.

**src/pages/Galery/Galery.jsx** → Page de la galerie photos/vidéos.

**src/pages/Home/Home.jsx** → Page d'accueil.

**src/pages/Storage/Storage.jsx** → Page du stockage.

---

**src/services/{service}.js** → Services permettant de contacter l'api (et interagir avec le socket du serveur).

**src/services/Pandora.js** → Contacter le service de la gestion du stockage en appelant la route /pandora.

**src/services/Photomaton.js** → Contacter le service de la gestion des albums via la route /photomaton.

**src/services/Report.js** → Contacter le service de la gestion des reports en appelant la route /report.

**src/services/Socket.js** → Se connecter au socket du serveur afin d'émettre, recevoir des informations sur la messagerie en temps réel.

**src/services/Users.js** → Contacter le service de la gestion des utilisateurs en appelant la route /users.

**src/services/Webfav.js** → Contacter le service de la gestion des sites favoris de l'utilisateur en appelant la route /webfav.

---

**src/main.jsx** → L'injection de la logique dans la balise #root.

**src/setupAxios.js** → Configuration d'axios.

## Commandes

<code>pnpm run start</code>	Démarrer le client
<code>pnpm run build</code>	Generer le fichier de build afin de le servir par le serveur. (manuel pour l'instant)