

Projet BitRuisseau

BitRuisseau

Mateen Salem Khalil
08/11/2024

Table des matières

Introduction	2
Analyse.....	2
Journal de travail	4
Etat des lieux	5
Démonstration du programme	6
Utilisation des fonctionnalités.....	6
Journal de travail	9
Conclusion	9
ChatGPT	10

Introduction

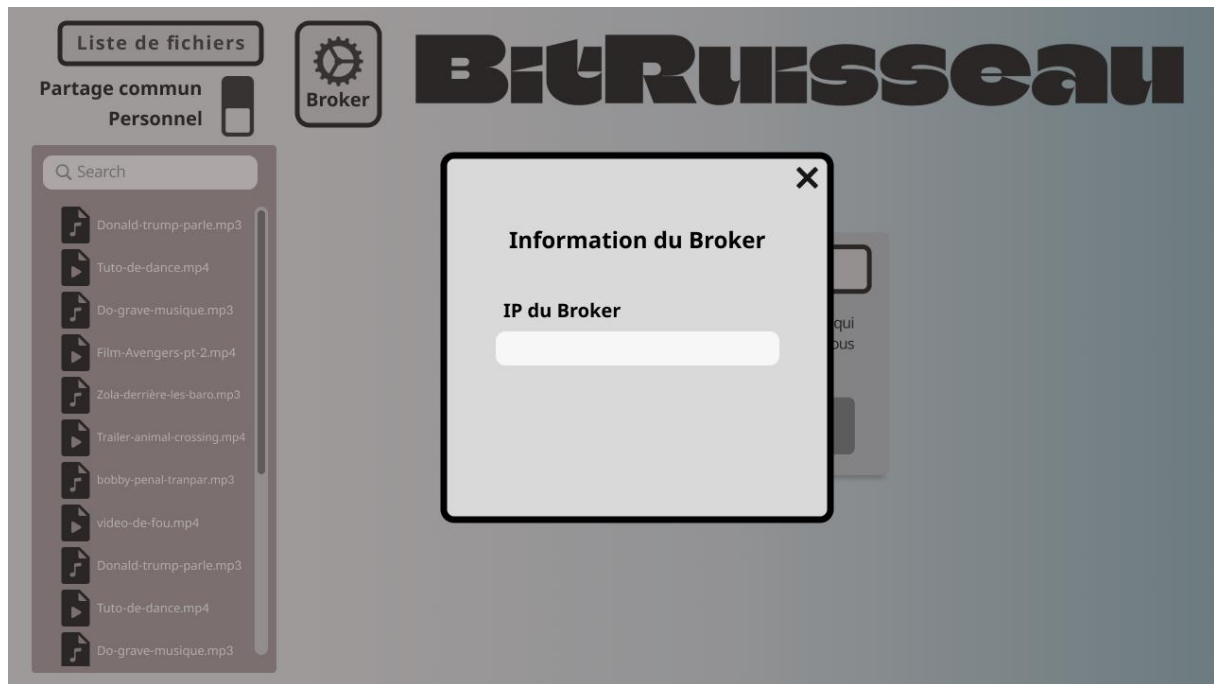
Dans le cadre de ce projet, nous avons développé une application mettant en œuvre les principes fondamentaux de la programmation distribuée, avec un accent particulier sur la communication inter-applications via le protocole MQTT. L'objectif principal était de créer un système capable de gérer l'échange d'informations concernant des catalogues multimédias, notamment des musiques, en facilitant à la fois le partage des métadonnées et des fichiers binaires correspondants.

Pour réaliser cet objectif, une architecture de communication normée a été mise en place. Cette architecture repose sur la définition de divers types de messages, chacun conçu pour répondre à des cas d'usage précis tels que la demande ou l'envoi de catalogues, ainsi que le transfert de fichiers. Les messages sont structurés de manière à garantir leur compréhension et leur efficacité dans un environnement réparti.

Analyse



Maquette 1



Maquette 2

Section	Élément	Fonctionnalité
Navigation	Liste de fichiers	Permet d'afficher la liste de tous les fichiers disponibles pour lecture et partage.
	Toggle partage commun/Personnel	Permet de basculer entre les fichiers partagés publiquement et les fichiers personnels.
	Broker (Réglages)	Ouvre un pop-up pour configurer l'adresse IP, port, username, password du broker MQTT.
	Barre de recherche	Permet de rechercher un fichier spécifique dans la liste de fichiers. (Spécifier comment elle fonctionne !!!!)
Gestion des Fichiers	Liste des fichiers	Affiche les fichiers disponibles sous forme de liste avec des icônes pour différencier les types (audio, vidéo). (Attributs des médias : type du fichier exemple : mp3, mp4, etc. Nom du fichier)
	Scrolling	Permet de naviguer entre les fichiers en défilant la liste.

	Double-clic fichier - Partage commun	Lorsqu'un fichier en partage commun est double-cliqué, il se télécharge dans le dossier localisé par l'utilisateur. Si aucun dossier n'est localisé, un pop-up s'affiche pour informer l'utilisateur qu'il doit d'abord sélectionner un dossier de stockage pour télécharger des fichiers partagés.
	Double-clic fichier - Local	Lorsqu'un fichier local est double-cliqué, il s'ouvre automatiquement pour lecture. Si le fichier est un fichier audio (ex. : mp3), il sera joué.
Dossier de Médias	Titre et description	Indique à l'utilisateur la fonction du dossier (contenant les fichiers à partager publiquement).
	Bouton « Localiser »	Permet de sélectionner le dossier qui contient les fichiers à partager avec les autres utilisateurs. Si dossier change, les médias partager ne seront plus indexer (fenêtre de confirmation quand on clique sur le bouton « Localiser » alors qu'un dossier l'est déjà
Configuration du Broker	Pop-up info du Broker	Affiche une fenêtre pour saisir l'adresse IP, port, username, password du broker MQTT, permettant la connexion pour le partage de fichiers.
	Fenêtre information "Information du Broker"	Permet à l'utilisateur de saisir l'adresse IP, port, username, password du broker MQTT pour établir la connexion nécessaire au partage et réception des fichiers.

Journal de travail

Etat des lieux

1. Fonctionnalités Complètes :

Localisation des Médias Locaux : L'application permet de localiser un dossier contenant les fichiers multimédias locaux et d'afficher la liste des fichiers disponibles.

Modification de la Configuration du Broker : Les utilisateurs peuvent modifier les paramètres de leur broker afin de se connecter aux autres utilisateurs.

Envoi et Récupération des Catalogues : L'utilisateur peut envoyer son catalogue de fichiers à partager et récupérer les catalogues des autres utilisateurs, facilitant ainsi la gestion des fichiers partagés.

Affichage des Médias Locaux et Partagés : Les fichiers locaux sont listés correctement, et les fichiers partagés dans les catalogues sont également affichés à l'utilisateur, offrant une vue d'ensemble des ressources disponibles.

2. Fonctionnalités Partiellement Fonctionnelles :

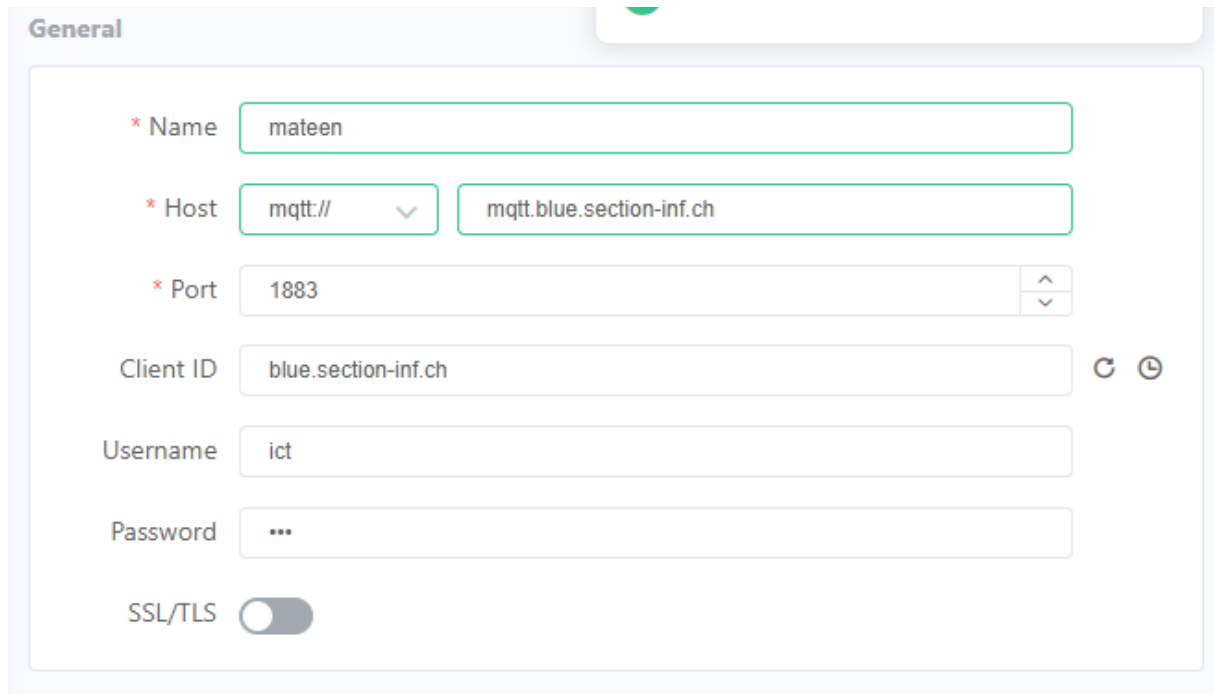
Demande de Téléchargement de Fichiers : L'utilisateur peut demander à télécharger un fichier à un autre utilisateur via le broker. Cependant, des erreurs techniques empêchent cette fonctionnalité d'être pleinement fonctionnelle. La demande de téléchargement échoue dans certains cas, et la logique de traitement des demandes reste instable.

Envoi de Fichiers Demandés : Bien que l'utilisateur puisse envoyer un fichier demandé par un autre utilisateur, cette fonctionnalité rencontre également des erreurs, notamment au niveau de la transmission des fichiers. Ces erreurs rendent l'envoi de fichiers partagés non fiable à ce stade.

Démonstration du programme

Configuration Mqtt :

Voici la configuration mqtt que j'ai utilisé pour mon programme



The image shows a screenshot of a software configuration window titled "General". It contains several input fields for MQTT settings:

- * Name:** A text field containing "mateen".
- * Host:** A dropdown menu showing "mqtt://" and a text field containing "mqtt.blue.section-inf.ch".
- * Port:** A text field containing "1883" with up and down arrow buttons.
- Client ID:** A text field containing "blue.section-inf.ch" with refresh and save icons.
- Username:** A text field containing "ict".
- Password:** A text field containing three dots "..." for masked input.
- SSL/TLS:** A toggle switch that is currently turned off.

Mon programme utilise par défaut le topic « test »

Utilisation des fonctionnalités

Envoie de catalogue :

Mon programme envoie son catalogue quand un émetteur le lui demande, ou bien quand l'utilisateur localise le dossier des médias

Topic: test QoS: 0

```
{"MessageType":1,"SenderId":"3dd08603-c6c4-4712-9cc9-10d5dbff801e"}
```

2025-01-15 22:28:36:489

Topic: test QoS: 0

```
{"MessageType":1,"SenderId":"3dd08603-c6c4-4712-9cc9-10d5dbff801e"}
```

2025-01-15 22:28:36:513

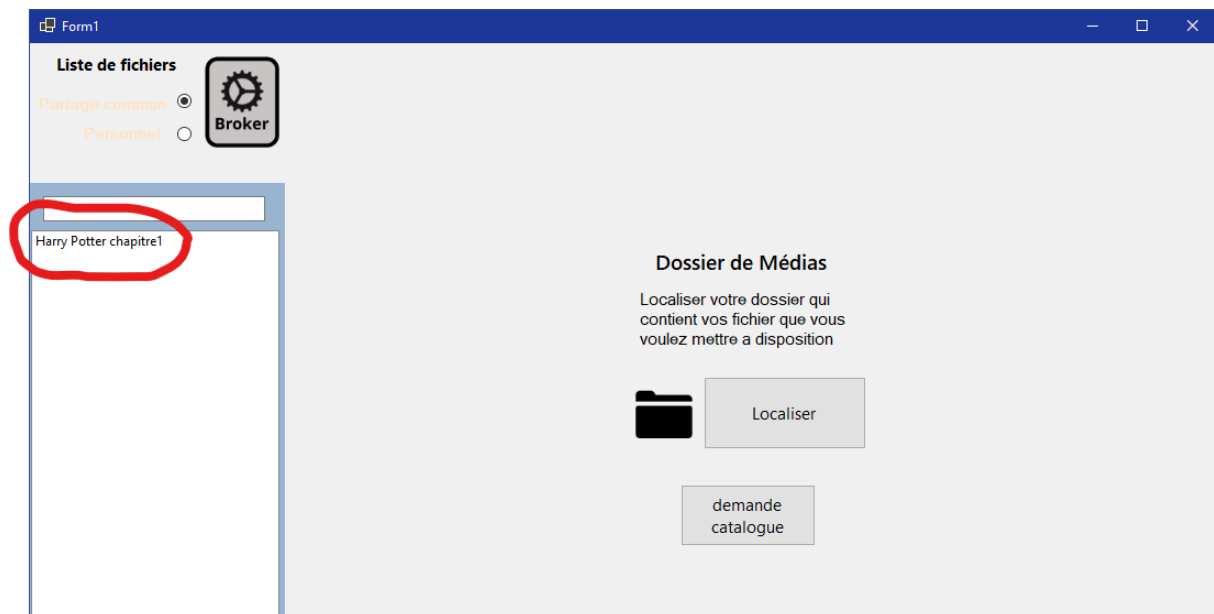
Topic: test QoS: 0

```
{
  "MessageType": 0,
  "SenderId": "85c3c255-bcc0-48fc-add6-812d52c89259",
  "EnveloppeJson": "{\r\n  \"Content\": [\r\n    {\r\n      \"Title\": \"IPLIKÉ140 - 14 HOODTRAP (remix par @75sleepy)\",\r\n      \"Artist\": \"Artiste inconnu\",\r\n      \"Type\": \"mp3\",\r\n      \"Size\": 8803403,\r\n      \"Duration\": \"00:03:40\",\r\n    },\r\n    {\r\n      \"Title\": \"C418_ Aria Math\",\r\n      \"Artist\": \"Artiste inconnu\",\r\n      \"Type\": \"mp3\",\r\n      \"Size\": 2764310,\r\n      \"Duration\": \"00:01:55\",\r\n    },\r\n    {\r\n      \"Title\": \"desktop\",\r\n      \"Artist\": \"Artiste Inconnu\",\r\n      \"Type\": \"ini\",\r\n      \"Size\": 452,\r\n      \"Duration\": \"Durée Inconnu\",\r\n    },\r\n    {\r\n      \"Title\": \"Theme of UNDER NINJA\",\r\n      \"Artist\": \"Shota Kowashi\",\r\n      \"Type\": \"mp3\",\r\n      \"Size\": 2382109,\r\n      \"Duration\": \"00:01:38\",\r\n    },\r\n    {\r\n      \"Title\": \"yt5s.com - Dream Speedrun Music (128 kbps)\",\r\n      \"Artist\": \"Artiste inconnu\",\r\n      \"Type\": \"mp3\",\r\n      \"Size\": 5107631,\r\n      \"Duration\": \"00:05:19\",\r\n    },\r\n    {\r\n      \"Title\": \"yt5s.com - Fly Me To The Moon (2008 Remastered) (128 kbps)\",\r\n      \"Artist\": \"Artiste inconnu\",\r\n      \"Type\": \"mp3\",\r\n      \"Size\": 2360879,\r\n      \"Duration\": \"00:02:27\"\r\n    }\r\n  ]\r\n}"
}
```

2025-01-15 22:28:36:580

Réception de catalogue :

Mon programme ajouter un media dans la liste file quand un nouveau catalogue est reçu :



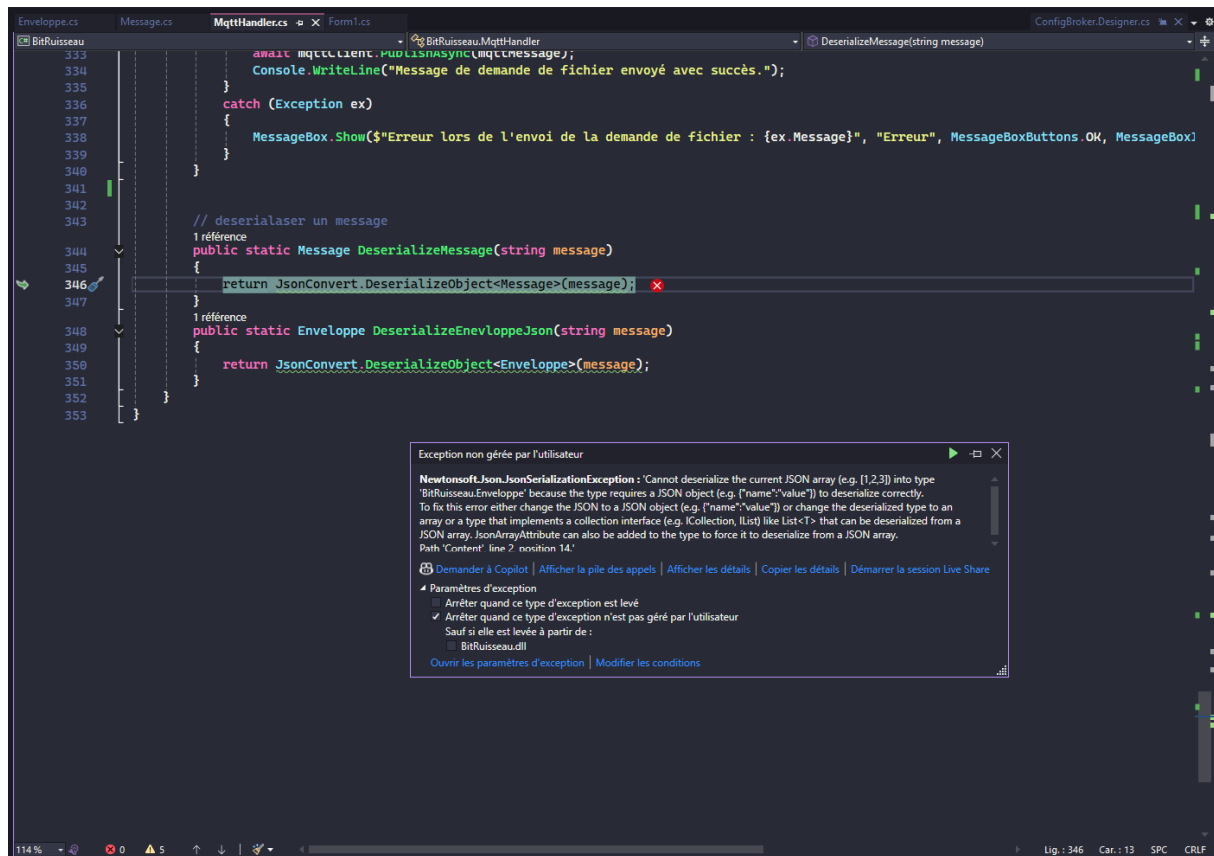
Demande de fichier :

Mon programme peut faire une demande fichier en simplement cliquant sur le fichier qu'on veut télécharger



Envoie de fichier :

Malheureusement c'est la seule fonctionnalité qui ne marche pas à 100%, quand mon programme reçoit une demande d'envoi de fichier il plante et affiche ce message d'erreur (voir ci-dessous).



Journal de travail

J'ai essayé de créer mon journal de travail avec gitjournal mais un message m'a indiqué que le nommage de mes commits n'était plus reconnu comme valide

Les commits suivants ont été ignorés, probablement parce qu'ils ne respectent pas la convention

feat(file_share) envoi et téléchargement de fichier WIP
 feat(recus de catalogue de désérialisation)
 feat(partage catalogue)_done et fonctionnelle
 feat(enveloppe catalogue)_ conforme aux autres élèves
 Revert 'essais code'
 feat(mqtt)_ ajout des classes FileRequest, Enveloppe, FileEnvelope, MediaData, bouton de demande de catalogue, ajout de méthode pour gerer les messages recus (pas fins), ajout de switch case pour décider quel action faire par rapport au type de message reçu
 feat(mqtt)_ ajout de types de messages (pas tous) et implémentation d'un gestionnaire MQTT pour les opérations sur les catalogues et les fichiers (pas fins non plus)
 (chore)_ refactor
 (fix) bug réglé localiser dossier
 feat(bouton config broker)_ début bouton config broker
 chore(bug)_ localisation dossier
 feat(mqtt)_ début de la logique pour utiliser mqtt
 feat(listes media)_ gestion listages de fichier, indexage dossier media
 feat(forms)_ logo, background
 feat(forms)_ background color
 chore(win-forms)_ création windows forms projet
 Doc(rapport)_ analyse fonctionnelle
 Doc(rapport)_ analyse fonctionnelle
 meeting (projet)
 meeting (projet)
 UX(forma)_ maquette
 Doc(Git)_projet, planification
 doc(rapport)_ titre, logo, font, sections
 meeting (projet)_ intro et explication du projet

Conclusion

Si ce projet devait être refait, je commencerais par éviter de tomber malade pour ne pas rater trois semaines de travail. Plus sérieusement, j'aurais aimé que la gestion du protocole commun soit plus claire et mieux définie dès le départ. La partie la plus difficile du projet pour moi a été de m'adapter au code déjà mis en place par les personnes ayant avancé plus rapidement.

Cela a été un défi de suivre leur rythme, surtout lorsque certaines décisions avaient été prises sans que tout le monde soit au courant ou aligné.

Cela dit, malgré ces difficultés, le projet a été une excellente opportunité d'apprentissage. L'expérience de travailler avec MQTT et de réaliser une analyse technique a été très enrichissante. J'ai aussi apprécié les aspects pratiques du projet, mais je pense que des améliorations dans la gestion du travail d'équipe et la définition des protocoles communs rendraient cette expérience encore plus fluide pour tout le monde.

ChatGPT

L'IA m'a été utile tout au long du projet, notamment pour comprendre la logique des codes de mes camarades et rendre leurs méthodes compatibles avec mon propre code. Elle m'a expliqué des segments de code complexes et m'a permis de mieux m'intégrer au projet en adaptant des méthodes pour qu'elles fonctionnent avec mon architecture. L'IA m'a aussi aidé à résoudre des erreurs en identifiant leurs causes et en suggérant des solutions, tout en m'offrant des conseils sur la gestion de la communication MQTT et la programmation distribuée.