

# Programowanie Obiektowe 2

---

## *Dokumentacja projektu*

### **Spis treści**

Aplikacja serwerowa .....	2
Aplikacja kliencka .....	3
Dodatkowe funkcjonalności .....	5
Otwieranie plików i tooltipy .....	5
System notyfikacji.....	6
Testy .....	7

## Aplikacja serwerowa

Okno aplikacji serwerowej zawiera widok drzewa plików znajdujących się na serwerze, listę aktywnych użytkowników, pasek statusu oraz okienko z logami.

Serwer przechowuje wszelkie dane w plikach w formacie .json, każdy z dysków przechowuje mapę, której kluczem jest plik a wartością lista posiadaczy tego pliku.

Lista użytkowników jest odświeżana przy każdym zalogowaniu i wylogowaniu się klienta natomiast drzewo plików gdy zostanie odebrany przez serwer nowy plik.

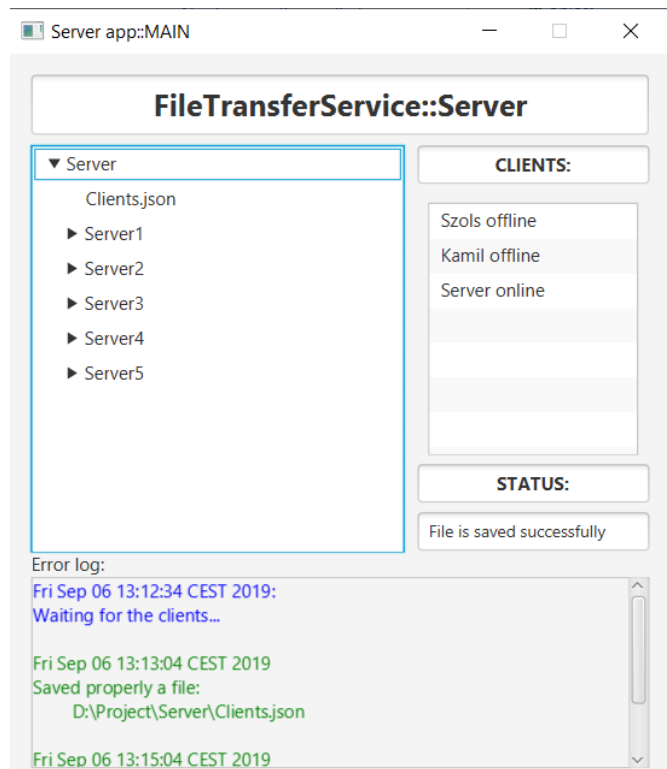
Lista aktywnych i nieaktywnych użytkowników jest wysyłana na żądanie klienta, gdy klient chciałby udostępnić jakiś plik użytkownikowi.

Serwer sprawdza czy są pliki dla nieaktywnych użytkowników, jeżeli są, wysyła im email z notyfikacją.

Każdy klient jest początkowo autoryzowany na sockecie autoryzacyjnym, jeżeli pomyślnie zostanie zalogowany, to otrzyma socket przeznaczony do komunikacji z przygotowanym wątkiem serwera i socket do wysyłania plików.

Proces komunikacji między klientem a wątkiem przebiega na podstawie komend, jeżeli serwer/klient chce przesłać pliki wysyła na początku żądanie do drugiego końca socketa w celu przygotowania obiektów klasy Runnable, które uruchamiają metody pobierające dane z socketa, Runnable są zakolejkowane przez pulę wątków zarówno przy wysyłaniu jak i odbieraniu plików przez klienta.

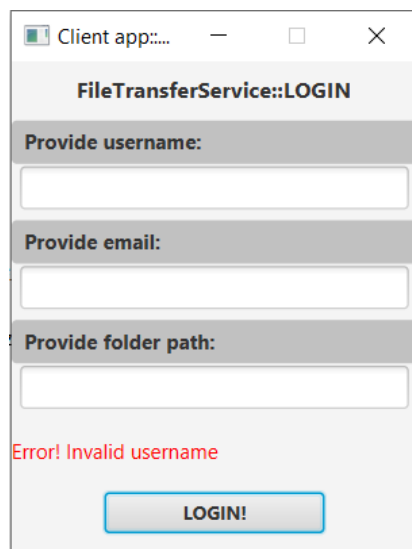
Serwer co dany czas dokonuje zapisu plików kontrolnych serwera – listy klientów, zawartości każdego dysku. Ponadto co minutę odświeża status na „Accepting clients...”.



## Aplikacja kliencka

Uruchomienie aplikacji klienckiej powoduje uruchomienie okna obok.

Użytkownik aby móc korzystać z serwisu musi podać swoje dane do logowania. Jeżeli poda niewłaściwe dane to wszystkie pola logowania zostaną wyczyszczone a użytkownik zostanie powiadomiony o błędzie stosownym komunikatem. Jak w przykładzie poniżej.

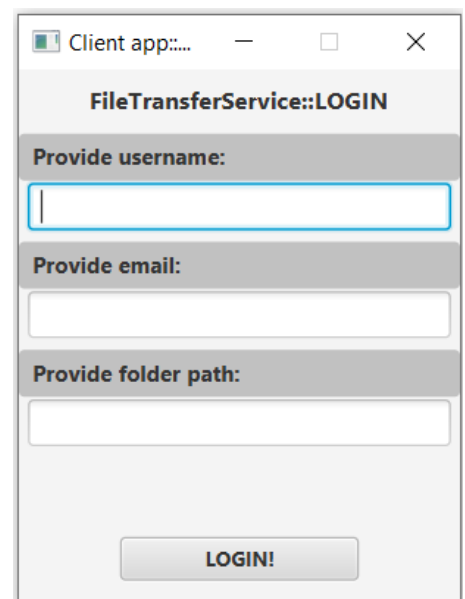
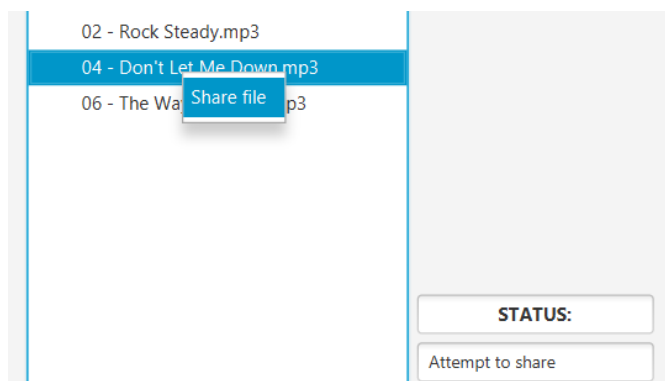


The screenshot shows a window titled "Client app:..." with a subtitle "FileTransferService::LOGIN". It contains three input fields labeled "Provide username:", "Provide email:", and "Provide folder path:". Below the fields is a red error message "Error! Invalid username" and a "LOGIN!" button.

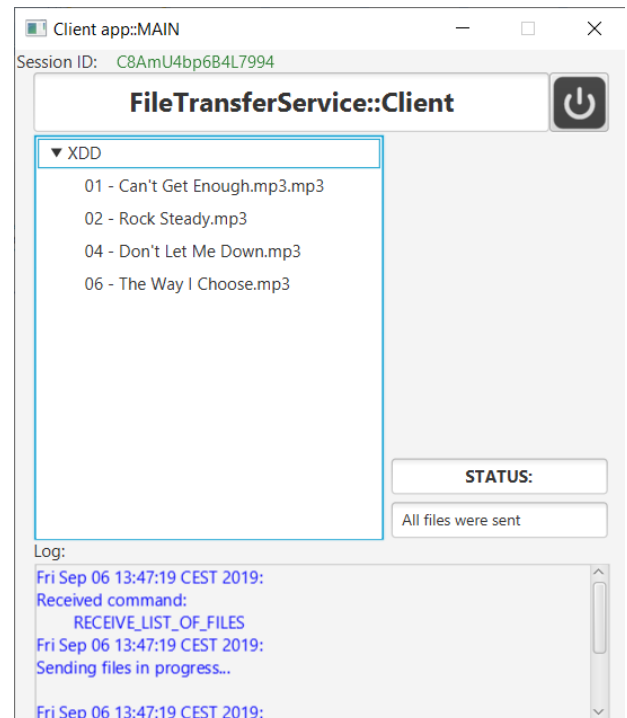
Jeżeli użytkownik zaloguje się poprawnie to przejdzie do głównego okna.

Główne okno składa się z widoku na drzewo plików użytkownika, paska statusu, okna logów, ID sesji, przycisku wylogowania się oraz niewidocznego widoku aktywnych i nieaktywnych klientów.

Jeżeli użytkownik kliknie prawym klawiszem myszy na plik, pojawi się menu „Share file”

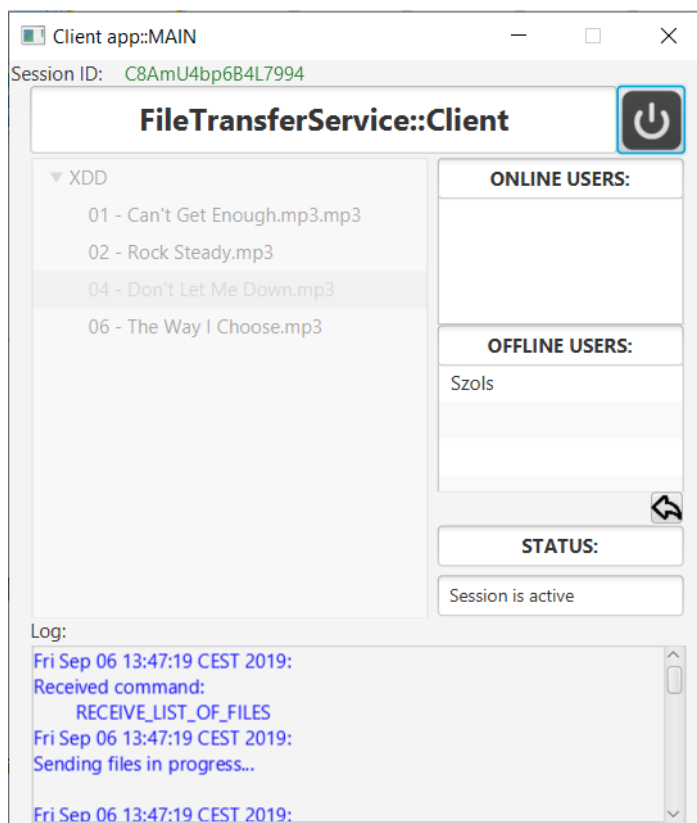


The screenshot shows a window titled "Client app:..." with a subtitle "FileTransferService::LOGIN". It contains three input fields labeled "Provide username:", "Provide email:", and "Provide folder path:". Below the fields is a "LOGIN!" button.



The screenshot shows a window titled "Client app:MAIN" with a subtitle "FileTransferService::Client". It displays a session ID "C8AmU4bp6B4L7994" and a list of files under the "XDD" folder: "01 - Can't Get Enough.mp3.mp3", "02 - Rock Steady.mp3", "04 - Don't Let Me Down.mp3", and "06 - The Way I Choose.mp3". There is a "STATUS:" label and a text box showing "All files were sent". A log section at the bottom shows the following text: "Log: Fri Sep 06 13:47:19 CEST 2019: Received command: RECEIVE\_LIST\_OF\_FILES Fri Sep 06 13:47:19 CEST 2019: Sending files in progress... Fri Sep 06 13:47:19 CEST 2019:". A power button is visible in the top right corner.

Jeżeli użytkownik kliknie w „Share file” to pojawi się widok list klientów:

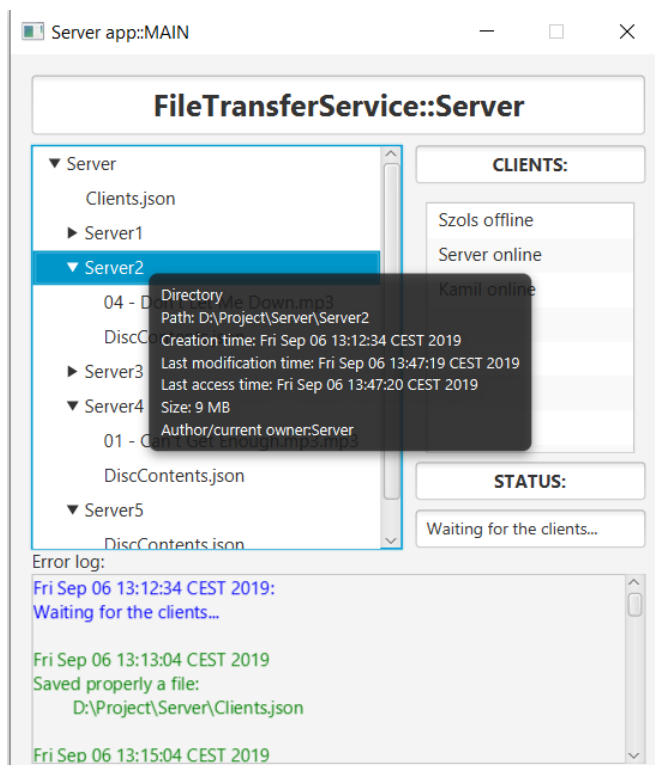


Widok list klientów umożliwia wybór użytkownika zarówno z listy on-line jak również listy off-line. Ponadto jest przycisk strzałki, który umożliwia powrót do widoku plików, jeżeli zdecydujemy się, że nikomu nie chcemy udostępnić plików.

## Dodatkowe funkcjonalności

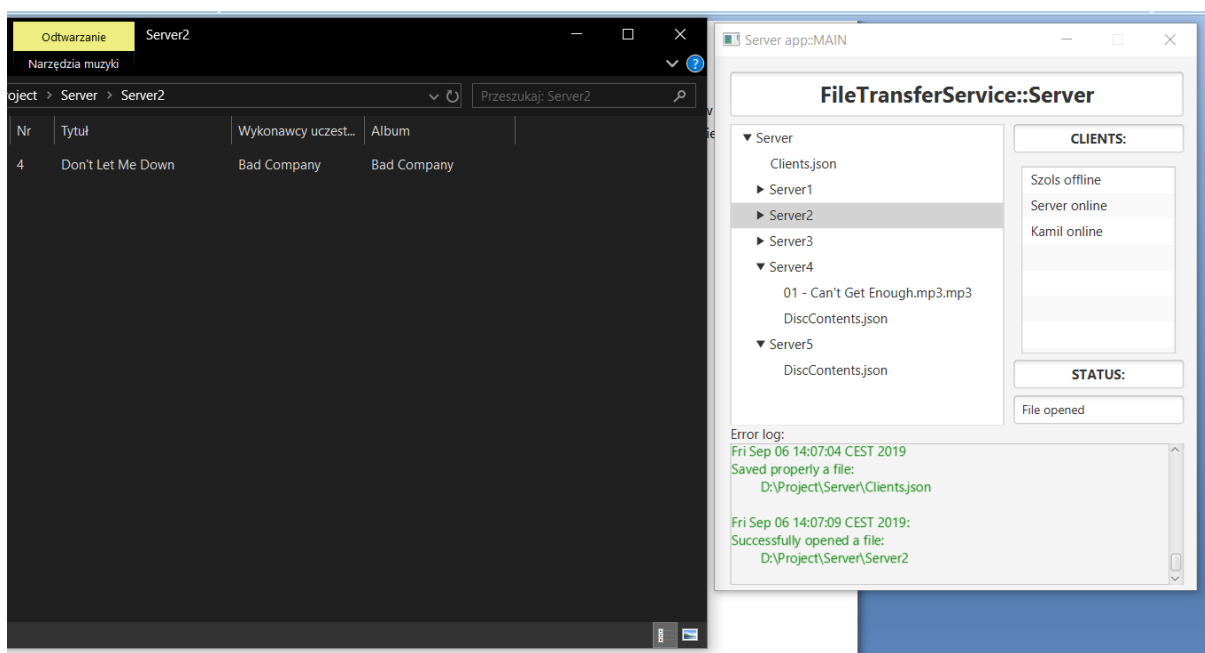
### Otwieranie plików i tooltipy

Zarówno Serwer jak i Klient mogą zobaczyć dokładny systemowy opis podstawowych atrybutów pliku jak również otworzyć posiadany plik za pomocą domyślnej aplikacji. Jeżeli najedziemy wskaźnikiem myszy na plik, bądź folder, zostanie wyświetlony tooltip pokazujący atrybuty pliku:



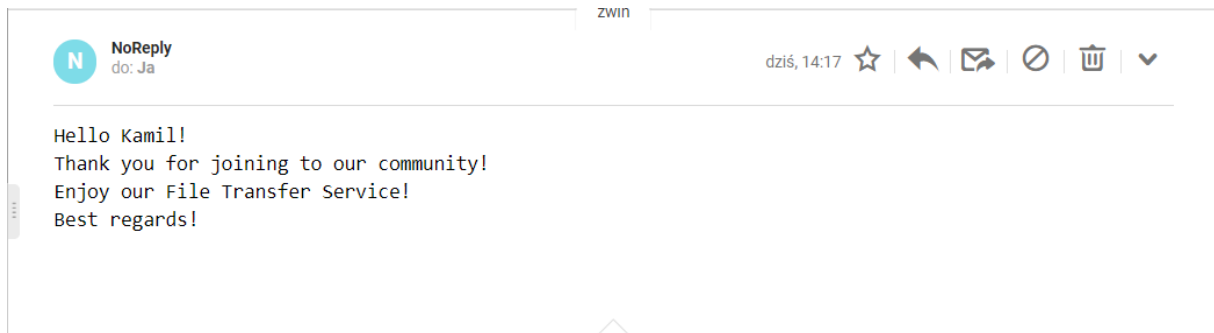
Rozmiar folderów jest obliczany poprzez przechodzenie wgłąb zawartości folderu.

Plik/Folder zostaje otwarty po dwukrotnym kliknięciu lewego przycisku myszy.



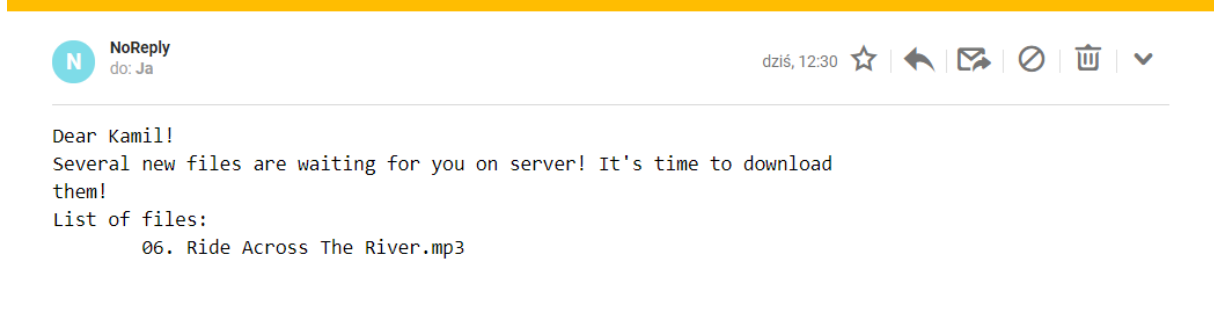
## System notyfikacji

Pierwsza notyfikacja pojawia się przy utworzeniu nowego konta, zostaje wysłany e-mail powitalny do użytkownika.

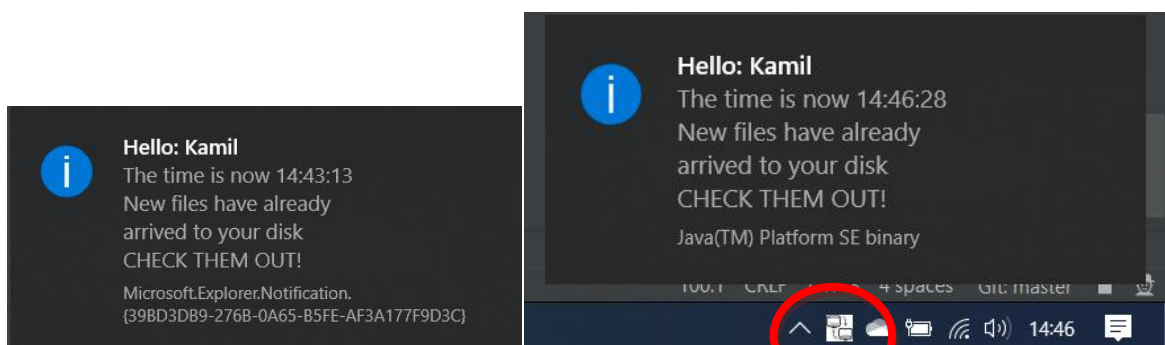


Jeżeli użytkownik jest offline, a ktoś udostępni mu plik w tym czasie, to otrzyma co pewien czas wiadomość, w której będzie znajdowało się powiadomienie o czekających nowych plikach.

## New files are waiting for you!



Ponadto gdy użytkownik jest on-line i otrzyma plik, gdy jego okno aplikacji jest zminimalizowane, to otrzyma powiadomienie systemowe:



Po kliknięciu myszką na tę ikonę zostanie przywrócony główny widok aplikacji klienta.

## Testy

Przeprowadzono 6 głównych testów, które badają najważniejsze funkcjonalności Klienta i Serwera. Przygotowane zostały klasy testowe pozwalające wykonywać operacje bez GUI z zachowaniem większości funkcjonalności.

Lista testów:

1. Sprawdzenie kodu błędu autoryzacji dla istniejącego użytkownika.
2. Sprawdzenie kodu błędu autoryzacji dla nowego użytkownika.
3. Sprawdzenie kodu błędu autoryzacji dla błędnego użytkownika.
4. Sprawdzenie komunikacji komendowej między poprawnie zalogowanym użytkownikiem a aplikacją serwera.
5. Sprawdzenie działania wątku obserwującego folder użytkownika oraz transferu wykrytego pliku do serwera.
6. Symulacja udostępnienia pliku użytkownikowi, wysłanie przez Serwer email-a do użytkownika, któremu udostępniamy plik (jest on off-line).

Wszystkie testy zostały pomyślnie uruchomione i zakończyły się sukcesem.