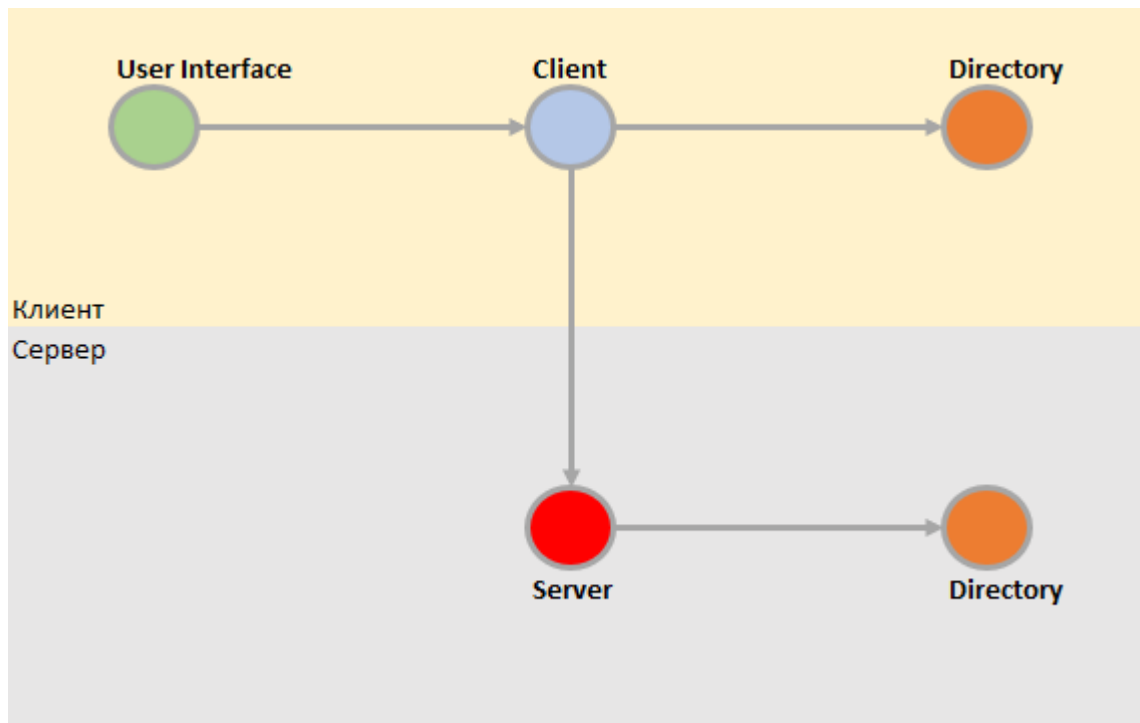


# Проект «Сетевое хранилище»

Концептуальная схема реализации:



Идея сводится к следующему:

1. Есть 4 отдельных модуля, каждый из которых выполняет свою набор функций в рамках построения общей цепи взаимодействий для выполнения общей задачи хранения файлов на сетевом ресурсе;

**User Interface** – форма(интерфейс) пользователя;

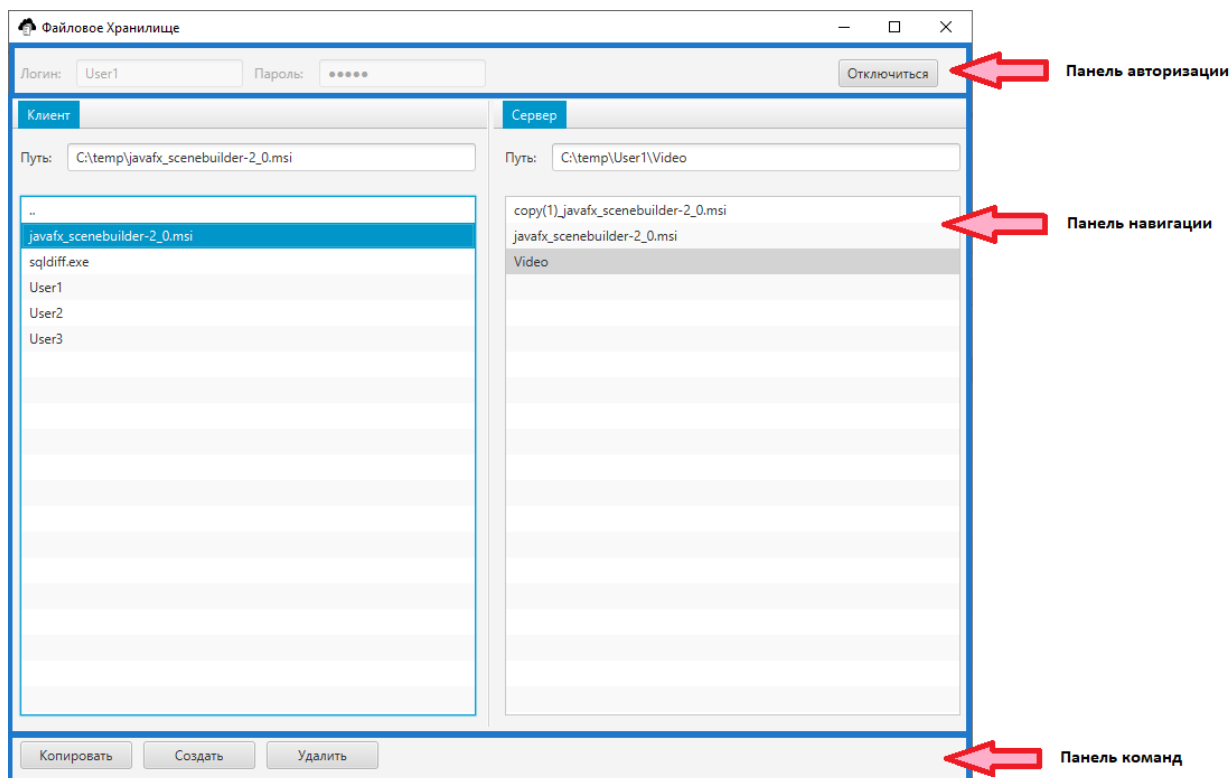
**Directory** – объект, соответствующий *текущей* директории, который хранит все результаты действий пользователя(список объектов, выбранный пользователем объект, его тип, ссылка на родительскую директорию и др.);

**Client** – объект, выполняющий основные функции по навигации пользователя по доступным директориям и непосредственно копирование файлов между пользователем и Сервером;

**Server** – объект, выполняющий те же функции, что и объект Client, только на Сервере(, т.е удаленном «хосте»).

2. Каждый модуль является полностью независимым(изолированным с т.з. разработки) объектом(функционалом) по отношению к другому модулю;
3. Все модули связаны между собой посредством *односторонних* связей, взаимодействуя друг с другом в режиме «запрос-ответ». В качестве протокола «общения» выбран формат JSON. Таким образом, каждый модуль имеет только один public-метод, который на «вход» получает запрос(от другого модуля), а на «выход» отдает готовый ответ(RESPONSE). Исключение составляет передача непосредственно данных копируемого файла.

## Интерфейс пользователя:



Интерфейс пользователя разделен на 3 части(панели):

**Панель авторизации** – выполняет ввод данных авторизации и, если требуется, есть возможность авторизоваться под другим пользователем, без разрыва соединения с Сервером.

**Панель навигации** – выполняется навигация по локальной файловой системе(Клиент) или по удаленной файловой системе Сервера, в пределах «стартовой» директории авторизованного пользователя. Плюс выбор файла для копирования(или удаления).

**Панель команд** – содержит кнопки для выполнения следующих команд:

«Копировать» - скопировать *файл* на Сервер или с Сервера в зависимости от того, какая из половин панели навигации активна:

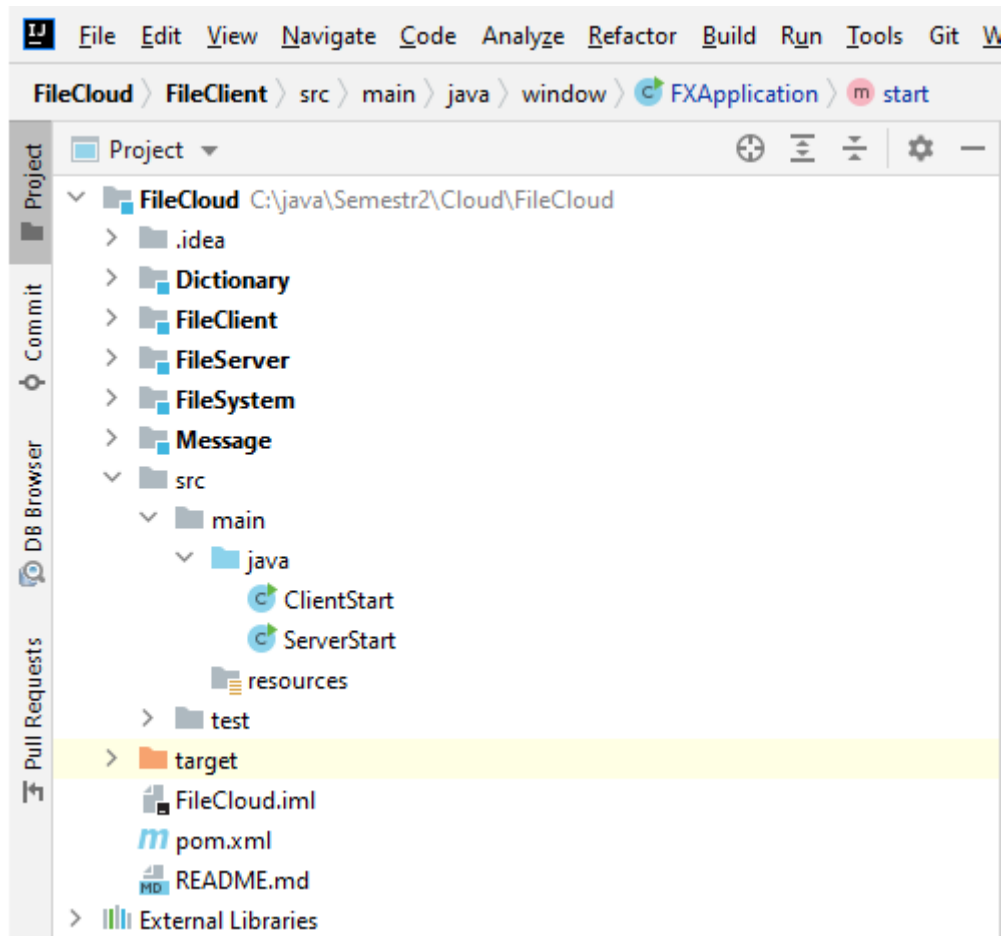
«Клиент» - файл копируется с Клиента на Сервер;

«Сервер» - файл копируется с Сервера на Клиент;

«Создать» - создать *директорию* в текущей директории Клиента или Сервера в зависимости от того, какая из половин панели навигации активна.

«Удалить» - удалить папку или файл в текущей директории Клиента или Сервера в зависимости от того, какая из половин панели навигации активна.

## Java реализация:



**FileCloud** – IDEA проект, содержащий функциональные модули реализации проекта;

**ClientStart** – запуск Клиента;

**ServerStart** – запуск Сервера.

**Dictionary** – модуль-справочник, содержащий исключительно java-перечисления, которыми являются:

**MessageTypes** – типы сообщений, которыми «общаются» между собой модули;

**CommandTypes** – типы команд, которые могут содержать сообщения;

**ResultCodes** – коды результата выполнения обработки сообщения;

**SelectTypes** – типы выбранного объекта в текущей директории («файл», «папка» и др.).

**Message** – модуль, реализации протокола «общения» между модулями:

**Mess** – класс, реализующий непосредственно само JSON-сообщение;

**MessUtil** – дополнительный класс, реализующий инструментарий работы с сообщениями.

**FileSystem** – модуль, с функционалом реализации модуля «Directory»(см. концептуальную схему)

**FileClient** – модуль, с функционалом Клиента:

**main** – пакет, с функционалом реализации модуля «Client»(см. концептуальную схему);

**window** - пакет, с функционалом реализации модуля «User Interface»(см. концептуальную схему).

**ServerClient** - модуль, с функционалом реализации модуля «Server»(см. концептуальную схему)

**ClientHandler** – класс-экземпляр Клиента на стороне Сервера.