Drop-Box LIKE

Project Name:	DB-Like
Group Number:	6
Group Members:	
	Ibrahim Muhammad
	Jaspreet S. Thind
	Mohsin Yasin

Project Setup:

Three instances of Amazon EC2 servers have been setup on Amazon.

Cirrus 0:
http://ec2-54-227-91-115.compute-1.amazonaws.com:31588
Listens on port 31589 for other server connections

Cirrus 1:
http://ec2-54-225-24-145.compute-1.amazonaws.com:31588
Listens on port 31589 for other server connections

Cirrus 2:
http://ec2-50-16-90-71.compute-1.amazonaws.com:31588
Listens on port 31589 for other server connections

To run a client instance:

- Change the client.json file (in config directory) and enter the relative path of the directory that you want to sync.
- Enter username and password for a user present in the users.json file.
- Do not modify anything else please.
- Once the server instances are up and running:
  (client side)
    - Navigate to /DBlike directory
    - Install required packages by using command prompt sudo npm install
    - Execute coffee client.coffee

Project Files:

::Configuration Directory::
CONFIG

On Client side the directory contains:

/client. json

- The path of the directory that needs to be synced.
- The login credentials of the user/client.
- List of all the servers (to simulate connection reset), and the server that the client is currently maintaining connection with.
- The save state of the connection.
- The timestamp when the directory was last synced

On Server side this directory contains:

/server.json

- The filestore that needs to be synced.
- List of the backup servers.
- The host and port information about the server

/users.json

- Contains credentials of all the users set up

::FILESTORE::

- Used by the servers to store and watch the directories of all the users.

::Node_Modules::

/auth.coffee

- Provides the basic authentication for the users privacy and data security.
- Authenticates user by comparing credentials to the ones stored with the server.
- On authentication provides user with a hash based token which the user can use to authenticate itself on and again during the current session.

/client.coffee

- Provides most of the client view functionality such as connection, disconnection and error scenarios.
- On connection, initiates a client_synchronizer object and a watcher object for every user session.

/server.coffee

- Provides most of the server view functionality such as connection, disconnection and error scenarios between server-client and server-server.
- On connection with user, initiates a server_synchronizer object and a server_watcher object for every user session.
- On connection with server, initiates a p2p_synchronizer object and a p2p_watcher object for every server session.

/wathcher.coffee
/server_watcher.coffee
/p2p_watcher.coffee

- The watcher watches for changes in the user specified directory (or filestore), and in case of any changes, sends information to the synchronizer about the change.

/client_synchronizer.coffee
/server_synchronizer.coffee
/p2p_synchronizer.coffee

- The synchronizer acts on the signal and emits an event to the Server, client or other server that a file has been updated and that the other app needs to download the file.

/common.coffee

- Includes all of the common packages required for the function of the project.

/package.json

- Includes the version of the packages that needs to be provided for a package manager install.

/util.coffee

- All the common utility functions needs by the program files.