

Video Server
Due Dec 13

Design and implement a concurrent server in Java that implements the protocol described in the Project Description. You need to implement the following commands:

login
courseList
videoList
questionList
questionAdd
answerList
answerAdd
logout

The server needs to work with your client developed in assignment 4. Any client that implements the project protocol should work with your server. The server does not have to implement the nonce, safeLogin and reset commands. (This does mean that you will have to modify your client to use login rather than safeLogin.) The server does not have to handle whitespace before or after the special characters as does the course server.

It is not safe to store passwords in plain text on the server. Each user has a password and a salt. Compute the MD5 hash of the password + salt and store that in the database. When the server receives the user name and password, the server first computes the MD5 hash of the password + salt and compares it to the value in the database. A list of usernames, passwords and salts are given in separate files. You will find links to them on the assignment page.

The server is to use your course PostgreSQL database account (which were handed out in class) to store data. The server needs to handle multiple requests at the same time. The server needs to log each client access to the server. The server uses several parameters such as: port to listen on, time out for sockets, database information (host, port, username, password). The server needs to have default values for the port and the timeout. The server needs to read a configuration file for all parameters. The parameters in the configuration file override the default values. The server also reads the command line to values to the parameters. The command line values override the ones in the configuration file and the default values.

Grading

Items	% of Grade
login courseList videoList questionList questionAdd answerList answerAdd logout	50
Concurrency	10
Config file & Command line	10
Runnable jar file	10
Using database	10
Quality of Code/Architecture	10

Milestones

You will not be turning in these milestones. They are just for your use personal use.

Milestone 1 Nov 25. Implement an iterative server (handles only one connection at a time) that logs connections, uses parameters from a config file and command line and implements the login and logout command. The login command can be faked by always returning 'ok:success'.

Milestone 2 Dec 2. Convert the iterative server to a concurrent server. Implement the login command to access the database to get the users password to check the user name and password are correct.

Milestone 3 Dec 9. Implement courseList, videoList and questionAdd using the database.

Final project Dec 13. Server completely implemented.

What to turn in

You need to turn in your complete eclipse project for your server. You also need to turn in a standalone jar file containing your run-able server. With the correct command one needs to be able to run your server using the jar file. Along with the jar file you need you configuration file and a readme file. The readme file needs to explain how to run your server using the jar file, what are the command line options, the parameters that can be set in your configuration file and anything else a user would need to know to run the server. The readme file, jar file and configuration file should be in a separate directory.