

source code: <https://github.com/kaillut/rpcproject>

used language:

python

used libraries:

rpyc (requires pip install)

requests (requires pip install)

sys

xml.etree.ElementTree

datetime

the client can do the following:

- ask user for input and send it to server
- get the content from server
- look for data on Wikipedia (links)

the server can do the following:

- give a list of topics
- give contents of topic
- add notes
 - o if topic is already created add a note to the end of it
 - o if no topic is found creates new one and adds note to it
 - o when note is added it searches the Wikipedia with the name of the note and adds link to the note if found
- read and save data to xml (the xml write doesn't include new lines, but it is valid xml (just harder to read))

installation:

install python, rpyc and requests

download source code from github

server.py requires db.xml (with atleast one node) in running folder

security: rpyc with the settings set in the code allows access to attributes and such so it isn't really secure, also the elements that the rpyc gives to the client can be modified from the client and so custom client can change any information on the server.

scalability: the server could with modifications could communicate changes in the data across multiple servers and be scalable, but currently the clients are only limited by the performance of the server

error handling:

almost no errors are handled on server side only if for some reason Wikipedia search fails, this is because the rpyc uses error propagation which causes the server-side errors propagate to the client and thus not affecting the servers operation and only crashing the client.

also on the client-side errors are not handled as this only is a showcase and it only affects client and thus not as important as server-side errors

transparency isn't really achieved as the client asks for connection information.

openness is somewhat achieved as the server doesn't check for what the client does and only does basic operations: adding and showing information.