Appendix I: main_wrapper.py

05/12/19 05:08:21 C:\GitHub\basic_blockchain\main_wrapper.py

```python
from html5lib import *
from blockchain_utils import *
import os

TABLE_LABELS = ['Proof', 'Next Block', 'Update Body', 'Time']
SEED_LENGTH = 256

# adds a new block to the webpage
def append_update(blockchain, proof, update_file, private_mode):
    with open(update_file, "r") as rd_file:
        update_block = rd_file.read()
        if private_mode:
            salt = os.urandom(SEED_LENGTH)
            update_block = salt.hex() + ' ' + HASH_FN(salt +
str.encode(update_block)).hexdigest()
        blockchain.append_block(proof, blockchain.tail, update_block)


# reads in information from an existing chain
def parse_chain(chain_folder, chain_head):
    if chain_folder[-1] != '/':
        chain_folder += '/'

    curr_block = chain_folder + chain_head

    with open(curr_block, "r") as ch_file:
        curr_block = ch_file.read()
        next_block = chain_folder + curr_block

    blocks = []

    while os.path.isfile(next_block):
        with open(next_block) as block_file:
            b1 = block_file.readline()
            b2 = block_file.readline()
            next_block = block_file.readline()[:-1]
            b3 = block_file.readline()
            b4 = block_file.readline()
        curr_block = (b1, b2, next_block + "\n", b3, b4)
        blocks += [curr_block]
        next_block = chain_folder + next_block
    return blocks

# uses existing blockchain files to generate the web page for that blockchain
def display(output_file, chain_folder, chain_head):
    blocks = parse_chain(chain_folder, chain_head)

    with open(output_file, 'w+') as out_file:
        out_file.write('<html><head><title>' + chain_folder + '</title>')
        out_file.write('<link rel="stylesheet" href="format.css"></head>')
        out_file.write('<body><h1 align="center">' + chain_folder + '</h1>')

        for b in blocks:
            out_file.write('<table>')
            out_file.write('<tr><th colspan=2>' + b[0] + '</th></tr> <col width="20%"/><col width="80%"/>')
            for i in range(0,4):
                out_file.write('<tr><td><b>' + TABLE_LABELS[i] + '</b></td><td>' + b[i+1] + '</td>')
            out_file.write('</table>')
            out_file.write('<br>')

        out_file.write('</body></html>')


# Example test code

# Generates the raw public version of the blockchain, with updates in plain text.
test_blockchain = Chain('test_blockchain', 512)
append_update(test_blockchain, 'proof1', 'test_blockchain_updates/update1', False)
append_update(test_blockchain, 'proof2', 'test_blockchain_updates/update2', False)
append_update(test_blockchain, 'proof3', 'test_blockchain_updates/update3', False)
append_update(test_blockchain, 'proof4', 'test_blockchain_updates/update4', False)
append_update(test_blockchain, 'proof5', 'test_blockchain_updates/update5', False)
display('test_blockchain.html', 'test_blockchain', 'test_blockchain')

# Generates the private version of the blockchain, with a salted and hashed version of the update, to protect proprietary code.
test_blockchain_private = Chain('test_blockchain_private', 512)
append_update(test_blockchain_private, 'proof1', 'test_blockchain_updates/update1', True)
append_update(test_blockchain_private, 'proof2', 'test_blockchain_updates/update2', True)
append_update(test_blockchain_private, 'proof3', 'test_blockchain_updates/update3', True)
append_update(test_blockchain_private, 'proof4', 'test_blockchain_updates/update4', True)
append_update(test_blockchain_private, 'proof5', 'test_blockchain_updates/update5', True)
display('test_blockchain_private.html', 'test_blockchain_private', 'test_blockchain_private')
```