НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Звіти до комп’ютерних практикумів з кредитного модуля “Технологія Блокчейн”

Прийняв

доцент кафедри

Яланецький В. А.

Виконав

Студент групи ІТ-02

Макаров І.С.

Київ – 2022

Комп‘ютерний практикум No 4

Мета:

Організація консенсусу в прототипі блокчейну

ВИКОНАННЯ

https://github.com/kinfi4/Kin-Blockchain

Почну з того, що наведу скріншот структури проекту, що вийшов в мене в кінці кінців:

Тут одразу такой допишу дисклеймер: у вас в завданні написано, що всі методи та класи повинні починатись з ПІБ студента, що максимально дивно, в мене стоїть на PyCharm плагін, що показує ім’я того, хто написав кожен клас та метод (плагін бере цю інфу з git), так я думаю буде і комфортно читати, і у вас буде доказ, що код написаний мною.

Ендпоінт призначенний для регестрації нових вузлів

```python
@router.post('')
@inject
def register_node(
    nodes_url: list[str],
    blockchain: Blockchain = Depends(Provide[Container.blockchain]),
):
    blockchain.register_node(nodes_url)

    return JSONResponse(status_code=status.HTTP_200_OK, content=f'All nodes were registered successfully!')
```

Ендпоінт для резолву конфліктів

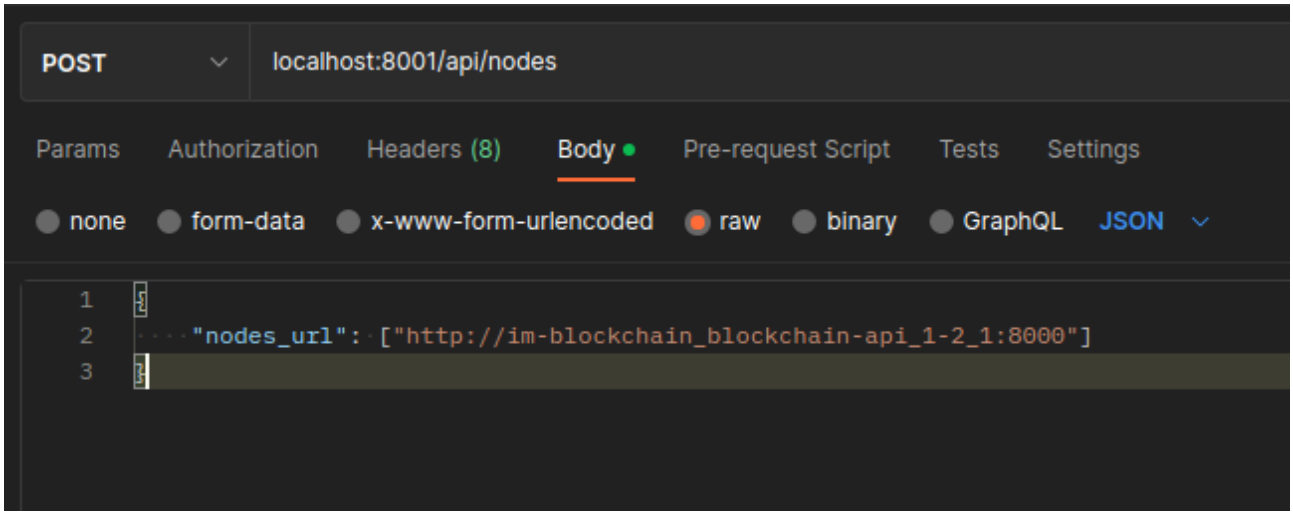
```python
@router.get('/resolve-conflicts')
@inject
def register_node(
    blockchain: Blockchain = Depends(Provide[Container.blockchain]),
):
    blockchain_is_replaced = blockchain.resolve_conflicts()

    message = 'Our blockchain is authoritative!' if not blockchain_is_replaced else 'Our blockchain was replaced!'

    return JSONResponse(
        status_code=status.HTTP_200_OK,
        content=message,
    )
```

Та відповідний код в самому сервісі блокчейну

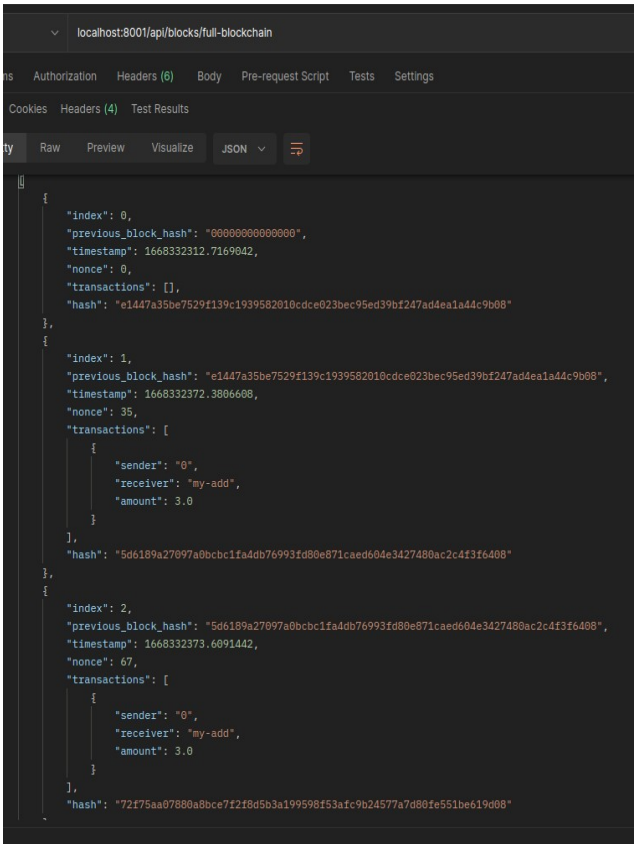
```
new *
def resolve_conflicts(self) -> bool:
    max_chain_length = len(self._bl_service.get_blockchain())
    exchanged_blockchain = False

    for host in self._registered_nodes:
        blockchain_url = f'http://{host}/api/blocks/full-blockchain'
        blocks = requests.get(blockchain_url).json()

        block_entities = [BlockEntity.from_dict(block) for block in blocks]

        if len(blocks) > max_chain_length and self._is_chain_valid(block_entities):
            self._bl_service.set_blockchain(block_entities)
            max_chain_length = len(blocks)
            exchanged_blockchain = True

    return exchanged_blockchain
```

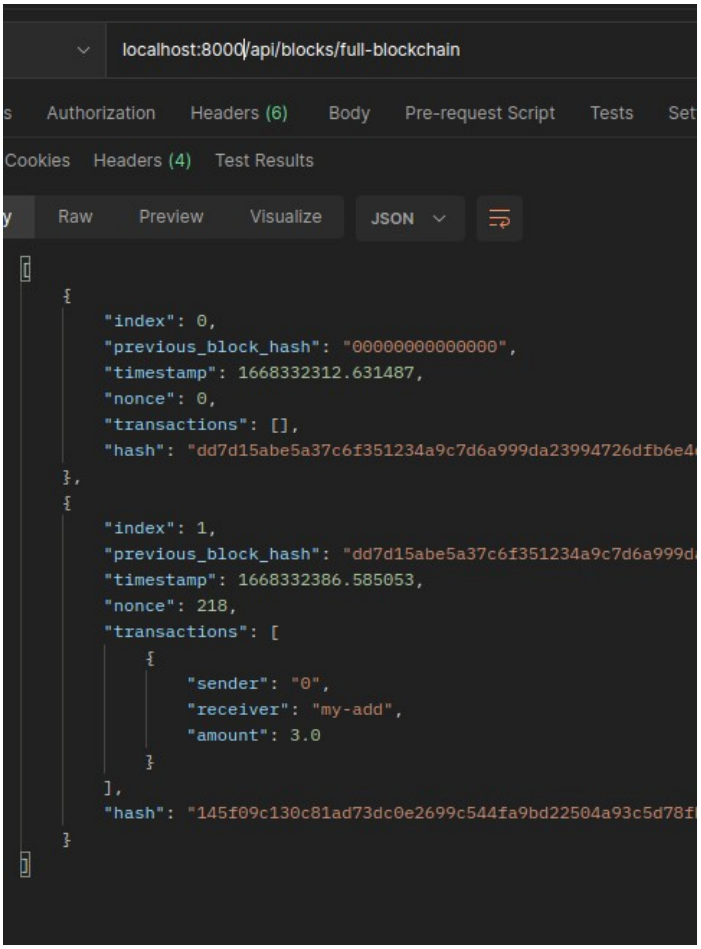
```
new *
def register_node(self, nodes_url: list[str]):
    for node_url in nodes_url:
        parsed_url = urlparse(node_url).hostname
        self._registered_nodes.add(parsed_url)

new *
def _is_chain_valid(self, blockchain: list[BlockEntity]) -> bool:
    new *
    def _is_nonce_valid(block: BlockEntity, nonce: int) -> bool:
        return block.nonce == nonce and block.get_hash()[-2:] == "08"

    for block_idx in range(1, len(blockchain)):
        if blockchain[block_idx - 1].get_hash() != blockchain[block_idx].previous_block_hash:
            return False

        if not _is_nonce_valid(blockchain[block_idx], blockchain[block_idx].nonce):
            return False

    return True
```

Давайте потикаємо апішку тепер. Запустимо дві ноди нашого блокчейну в докері, та зарейєструємо їх один у одного.

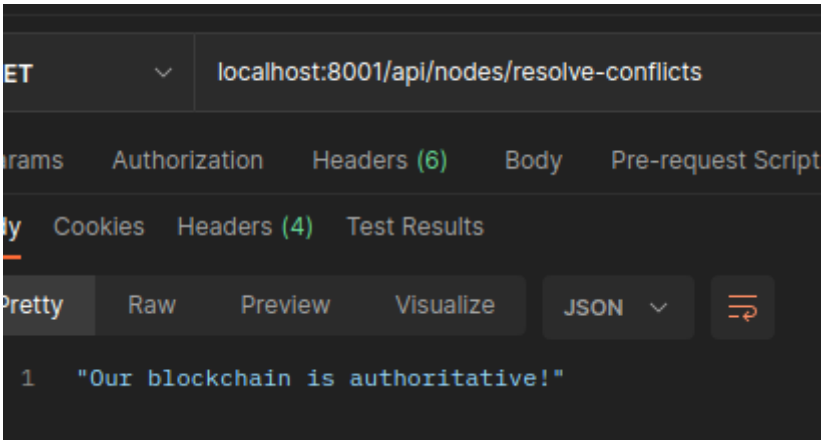


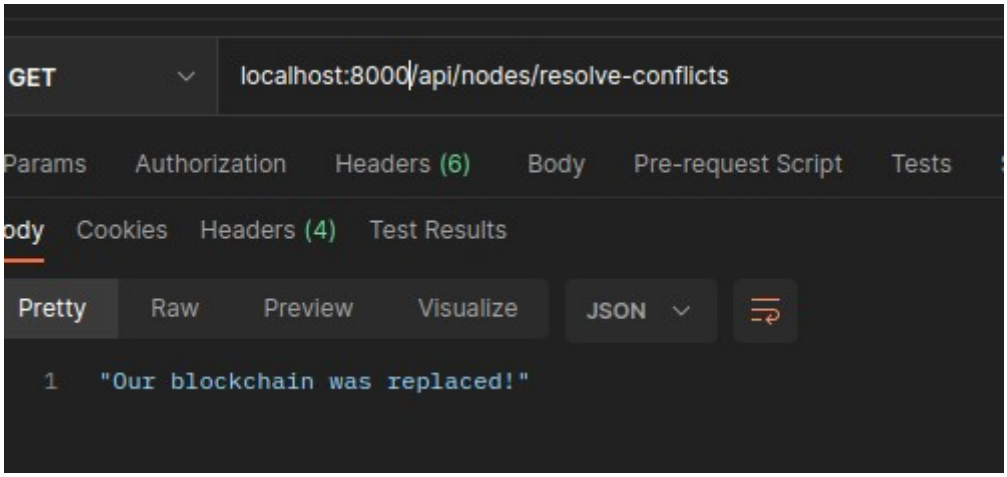
Тепер змайнемо два блоки у ноді, що на порті 8001 та один, що на порті 8000, таким чином ось наші блокчейни





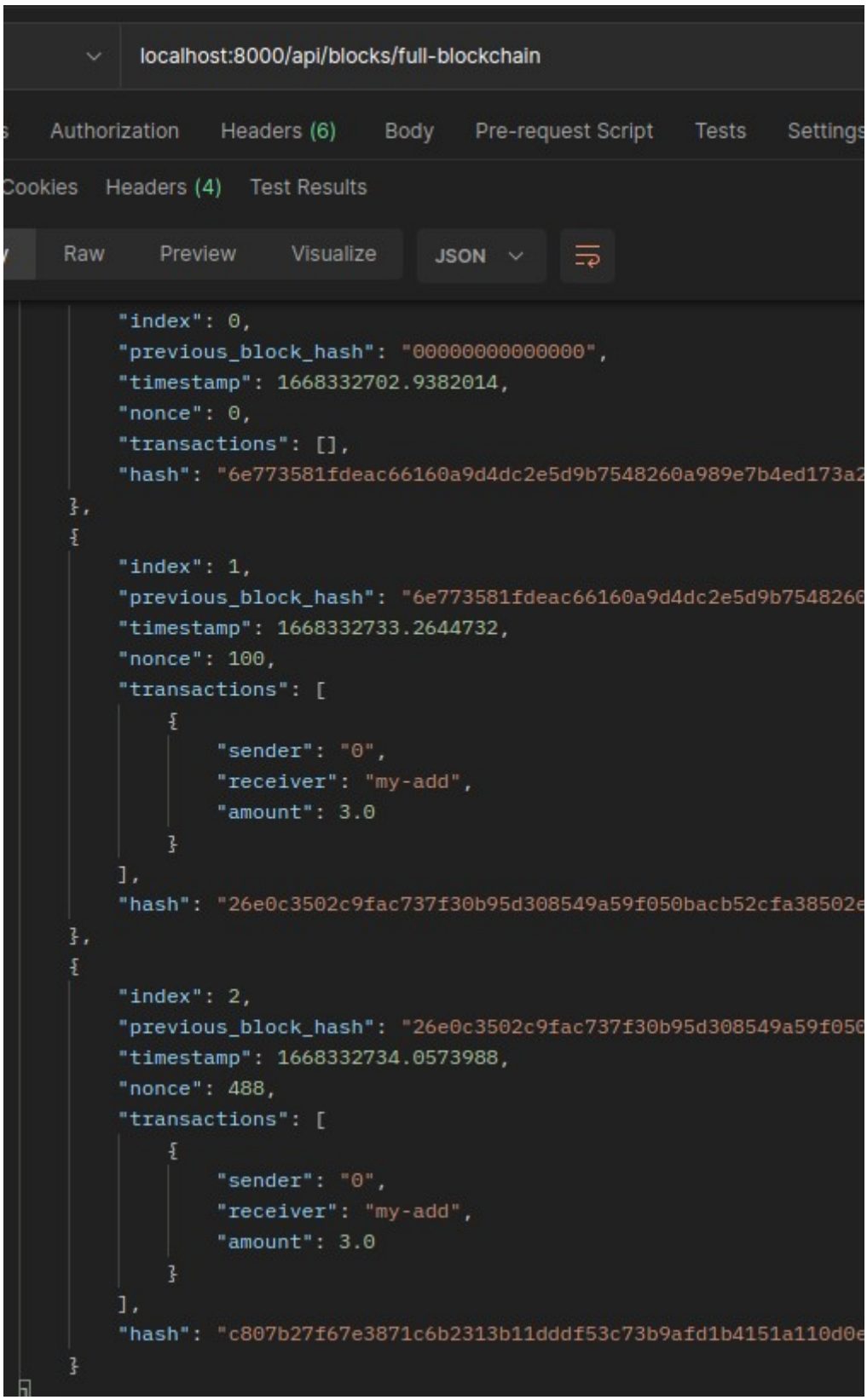
Як видно на 8001 порті блокчейн більше, давайте тепер спробуємо зарезолвети блокчени на обох нодах





Давайте тепер подивимось на наш блокчейн на 8000 порті.

Як видно тепер він став довше, а саме замінився тим, що був у нас на 8001


localhost:8000/api/blocks/full-blockchain
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Headers (4)
Test Results
Raw
Preview
Visualize
JSON
"index": 0,
"previous_block_hash": "00000000000000",
"timestamp": 1668332702.9382014,
"nonce": 0,
"transactions": [],
"hash": "6e773581fdeac66160a9d4dc2e5d9b7548260a989e7b4ed173a2
},
{
"index": 1,
"previous_block_hash": "6e773581fdeac66160a9d4dc2e5d9b754826
"timestamp": 1668332733.2644732,
"nonce": 100,
"transactions": [
{
"sender": "0",
"receiver": "my-add",
"amount": 3.0
}
],
"hash": "26e0c3502c9fac737f30b95d308549a59f050bacb52cfa38502e
},
{
"index": 2,
"previous_block_hash": "26e0c3502c9fac737f30b95d308549a59f050
"timestamp": 1668332734.0573988,
"nonce": 488,
"transactions": [
{
"sender": "0",
"receiver": "my-add",
"amount": 3.0
}
],
"hash": "c807b27f67e3871c6b2313b11dddf53c73b9afd1b4151a110d0e
}

ВИСНОВОК

В даній роботі, ми додали алгоритм консенсусу до нашого блокчейну.