

블록체인 (Blockchain)

04. 코인 거래 기능 구현 및 작동

소프트웨어 끝대 강의

노기섭 교수

(kafa46@cju.ac.kr)

강의 내용 정리 페이지 안내

- 강의는 주제별로 구분하여 github에 올려 놓았습니다.
 - ppt 슬라이드, 유튜브 동영상 링크, 소스코드
 - venv 폴더는 업로드하지 않습니다.
 - 개인별로 설치/사용 환경이 다르기 때문에...
 - 각자 requirements.txt 활용해 구성한 가상환경을 사용하면 됩니다.
 - URL: https://github.com/kafa46/cju_coin

Contents

4.1 거래 기능 담당 클래스 구현

- transfer.py

4.2 거래 기능 API 구현

main_views.py

def transactions()

- 거래 내역 요청 (GET)

- 거래 내역 추가 (POST)

4.3 UI 구축

/templates/index.html

/static/js/transfer.js

필요한 몇 가지 사전지식

Prerequisites

Bootstrap 소개



나무 위키: <https://namu.wiki/w/Bootstrap>

■ 개발자는 개발에 집중하자!

- 디자인은 디자이너에게!!!
- 대안: bootstrap 활용 (official docs: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>)

■ CDN 설정 (참고 문서: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>)

- 설정 방법: .html 파일에 아래 링크 삽입
 - `<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">`
 - `<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNnguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>`

■ 우리가 사용할 몇 가지 예제

- 버튼 buttons
 - <https://getbootstrap.com/docs/5.3/components/buttons/#base-class>
- 입력 input group
 - <https://getbootstrap.com/docs/5.3/forms/input-group/#basic-example>

jQuery 소개



■ 개발자의 삽질을 줄여주자!

- 길고 복잡한 JavaScript 코드를 간단하게 사용하자!!!

■ CDN 설정 (참고 문서: <https://releases.jquery.com/>)

- 설정 방법: .html 파일에 아래 링크 삽입
 - `<script src="https://code.jquery.com/jquery-3.7.0.js" integrity="sha256-JlqSTELeR4TLqP0OG9dxM7yDPqX1ox/HfgiSLBj8+kM=" crossorigin="anonymous"></script>`

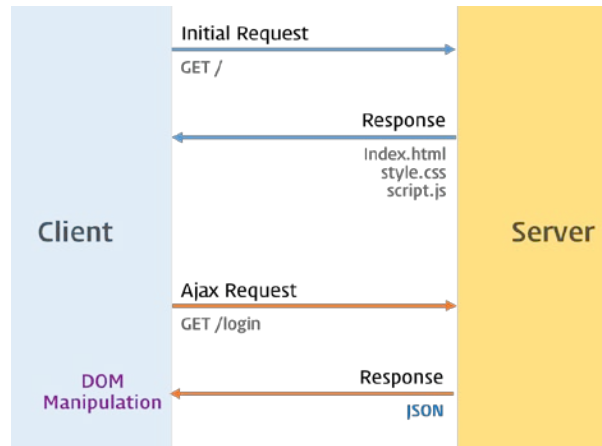
■ REST API 구축할 때 활용

- 요소를 선택
- 편리하게 JavaScript 제어

Ajax 소개

■ Ajax : Asynchronous JavaScript and XML

비동기 처리 Asynchronous



이미지 출처:
<https://velog.io/@binest03459/Ajax-Asynchronous-JavaScript-and-XML>

- 자바스크립트를 이용해 서버와 브라우저가 비동기 방식으로 데이터 교환
- Web에서 데이터를 조회, 생성, 업데이트, 삭제하고 싶을 경우,
 - 전체 페이지를 새로 고침 refresh 하지 않기 위해 사용
 - get, post, put, delete API 구현에 자주 활용

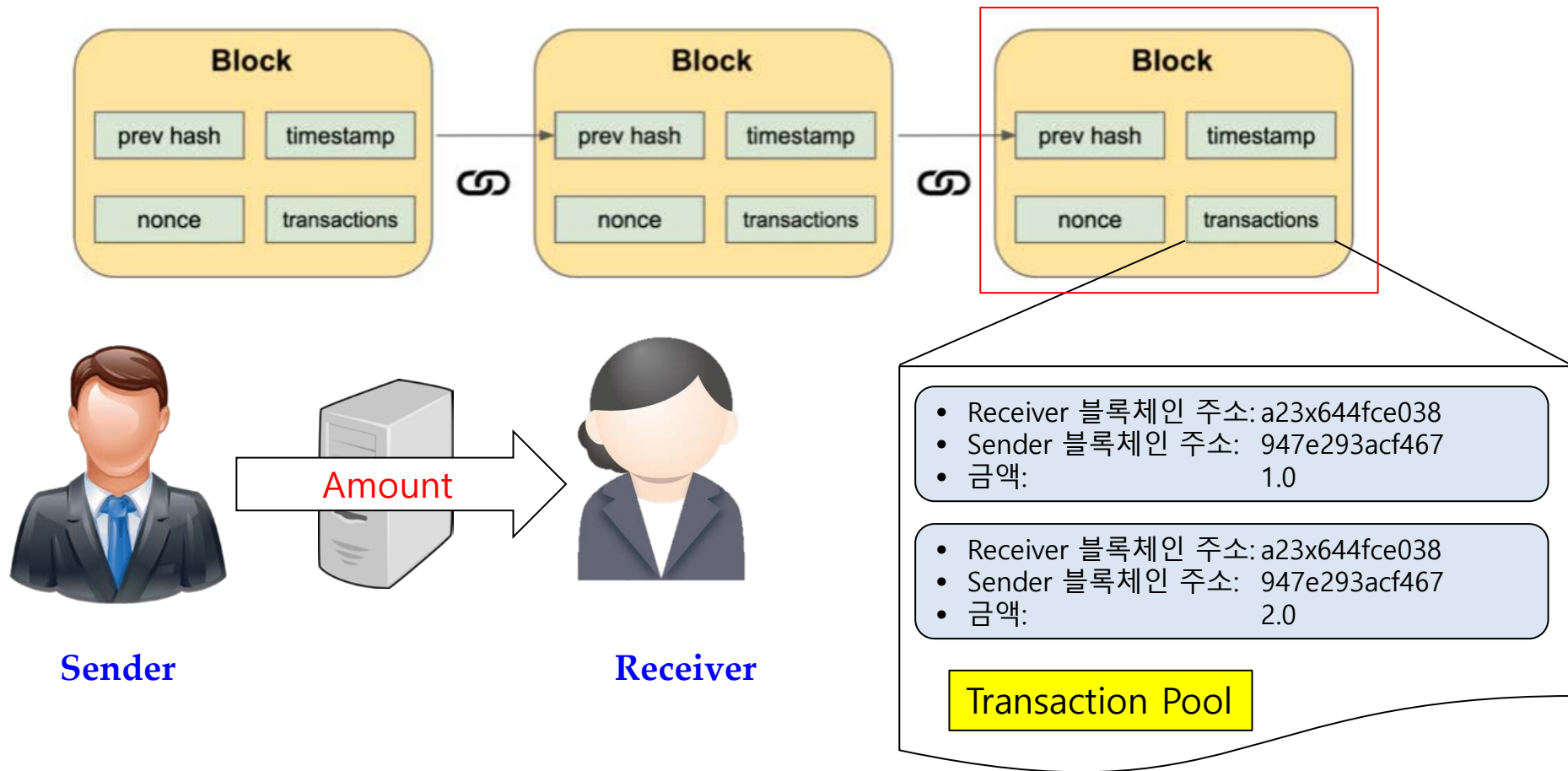
■ jQuery에서 완벽하게 지원

- jQuery만 있으면 됩니다.

거래 담당 클래스 구현

transfer.py

거래 (transaction) 추가하려는 상황



거래 담당 클래스 (transfer.py)

```
from mining import db
from mining.models import Block
from mining.models import Transaction
```

```
class Transfer:
    '''거래 담당 클래스'''
```

```
    def __init__(
        self,
        # send_public_key: str, <- Todo
        send_blockchain_addr: str,
        recv_blockchain_addr: str,
        amount: float,
        # signature: str = None <- Todo
    ) -> None:
```

```
    # self.send_public_key = send_public_key <- Todo
    self.send_blockchain_addr = send_blockchain_addr
    self.recv_blockchain_addr = recv_blockchain_addr
    self.amount = amount
    self.block_id = Block.query.filter(
        Block.timestamp,).order_by(Block.timestamp.desc()).first().id
    # self.signature = signature <- Todo
```

```
    def add_transaction(self) -> bool:
        '''Add a transaction into DB'''
        # Todo
        # 마이닝(채굴) 검증없이 transaction pool에 추가

        # Todo
        # transaction 검증 과정
        # is_verified = self.verify_transaction_signature()
        is_verified = True

        # 검증을 통과하면 DB에 추가
        if is_verified:
            self.commit_transaction()
            return True
        return False
```

컨트롤러 코딩 (main_views.py)

```
@bp.route('/transactions/', methods=['GET', 'POST'])
def transactions():
    '''Transaction.transaction_pool 정보를 읽어서 리턴'''
    block_chain = get_blockchain()
    if request.method=='GET':
        '''transaction 정보 요청'''
        print('Transaction 정보 제공')
        resp = {
            'transactions': block_chain.get('transaction_pool'),
            'length': len(block_chain.get('transaction_pool')),
        }
        return jsonify(resp), 200

    if request.method=='POST':
        '''transaction 추가'''
        print('블록체인 노드: transaction 추가')
        request_json = request.json
        print(f'request_json: {request_json}')
        transfer = Transfer(
            # send_public_key=request_json.get('send_public_key'), <- Todo
            send_blockchain_addr=request_json.get('send_blockchain_addr'),
            recv_blockchain_addr=request_json.get('recv_blockchain_addr'),
            amount=request_json.get('amount'),
            # signature=request_json.get('signature') <- Todo
        )
        is_transacted = transfer.add_transaction()

        if not is_transacted:
            return jsonify({'status': 'fail'}), 400
        return jsonify({'status': 'success'}), 201
```

User Interface 코딩 (index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>채굴</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJJOZ" crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYN57NTKfAdvQSZ" crossorigin="anonymous"></script>
  <script src="https://code.jquery.com/jquery-3.7.0.js" integrity="sha256-JlqStELEr4TLqP00G9dxM7yDPqX1ox/HfgiSLBj8+kM="
crossorigin="anonymous"></script>
</head>
<body>
  <script src="/static/js/transfer.js"></script>
  <div class="btn btn-outline-primary m-3" id="query_transactions">거래 내역 조회</div>
  <div class="row m-3" id="form-area">
    <form method="post" id="transfer-form">
      <div class="my-5">
        <label for="send_blockchain_addr" class="form-label">보내는 사람 지갑 주소</label>
        <input type="text" class="form-control"
          id="send_blockchain_addr" name="send_blockchain_addr"
          autoComplete="off"
          placeholder="받는사람 지갑 주소를 입력해주세요.">
      </div>
      <div class="my-5">
        <label for="recv_blockchain_addr" class="form-label">받는 사람 지갑 주소</label>
        <input type="text" class="form-control"
          id="recv_blockchain_addr" name="recv_blockchain_addr"
          autoComplete="off"
          placeholder="받는사람 지갑 주소를 입력해주세요.">
      </div>
      <div class="input-group mb-3">
        <div class="input-group-prepend">
          <span class="input-group-text" id="basic-addon3">
            이체 수량
          </span>
        </div>
        <input type="number" class="form-control" id="amount" name="amount"
          min="0.000000001"
          step="0.000000001"
          placeholder="최소 0 초과 코인 수량을 입력하세요(소수점 9자리까지 가능).">
      </div>
      <button class="btn btn-outline-primary" id="submit-btn">거래 추가</button>
    </form>
  </div>
</body>
</html>
```

Javascript를 이용한 API 연결 (transfer.js)

```
$(function(){

    // 거래내역 조회 (GET 요청)
    $('#query_transactions').on('click', function(){
        $.ajax({
            url: '/transactions',
            type: 'get',
            dataType: 'json',
            success: function(response){
                console.log(response.transactions)
                console.log(response.length)
            },
            error: function(error){
                console.log('에러가 발생했어요 ㅠㅠ', error)
            }
        });
    });

    // 거래내역 추가 (POST 요청)
    $('#submit-btn').on('click', function(e){
        e.preventDefault();
        let send_data = {
            'send_blockchain_addr': $('#send_blockchain_addr').val(),
            'recv_blockchain_addr': $('#recv_blockchain_addr').val(),
            'amount': $('#amount').val(),
        }
        $.ajax({
            url: '/transactions',
            type: 'post',
            data: JSON.stringify(send_data),
            contentType: 'application/json',
            dataType: 'json',
            success: function(response){
                if (response.status==='success'){
                    alert('거래 추가 성공!')
                }
            },
            error: function(error){
                alert('에러가 발생했어요 ㅠㅠ', error)
            }
        });
    });
});
```



다음 강의

→ 지갑 (wallet) 작성

수고하셨습니다 ..^^..