

# 블록체인 (Blockchain)

## 03. 개발환경 세팅, Blockchain 클래스 작성

소프트웨어 공대 강의

노기섭 교수

([kafa46@cju.ac.kr](mailto:kafa46@cju.ac.kr))

# 플라스크 설정 및 DB 생성

# 설정 파일 작성 (config.py)

```
import base64
import os
import requests

from mining.secret import csrf_token_secrete

BASE_DIR = os.path.dirname(__file__)

# SQLAlchemy_DATABASE_URI -> SQLite DB 접속 주소
SQLALCHEMY_DATABASE_URI = 'sqlite:///{}'.format( os.path.join(BASE_DIR, 'blockchain.db'))

# ORM 객체의 변경사항을 지속적으로 추적하고 변동 이벤트에 대한 메시지 출력
# 불필요한 경우 False로 꺼 놓는 것을 추천
SQLALCHEMY_TRACK_MODIFICATIONS = False

# Seret key for CSRF token
SECRET_KEY = csrf_token_secrete

# 채굴자에게 보내는 블록체인 네트워크 이름
BLOCKCHAIN_NETWORK = 'BLOCK CHAIN NETWORK'

# Mining Difficulty
MINING_DIFFICULTY = 5

# Mining Reward (채굴 보상금)
MINING_REWARD = 15.0
```

# 서버 초기화 파일 작성 (\_\_init\_\_.py)

```
from flask import Flask
from flask_migrate import Migrate
from flask_sqlalchemy import SQLAlchemy

from mining import config

migrate = Migrate()
db = SQLAlchemy()

def create_app():
    '''app 객체 생성'''
    app = Flask(__name__)
    app.config.from_object(config)

    # DB 초기화
    db.init_app(app)

    # Using Batch Mode: [SQL: ALTER TABLE user DROP COLUMN XX] 방지
    migrate.init_app(app, db, render_as_batch=True)

    # View import
    from mining.views import main_views

    app.register_blueprint(main_views.bp)

    # Return App
    return app
```

# 뷰(컨트롤러) 파일 작성 (/views/main\_views.py)

## ■ main\_views.py

```
from flask import (
    Blueprint,
    jsonify,
)

from mining.utils.blockchain_utils import get_blockchain

bp = Blueprint('main', __name__, url_prefix='/')

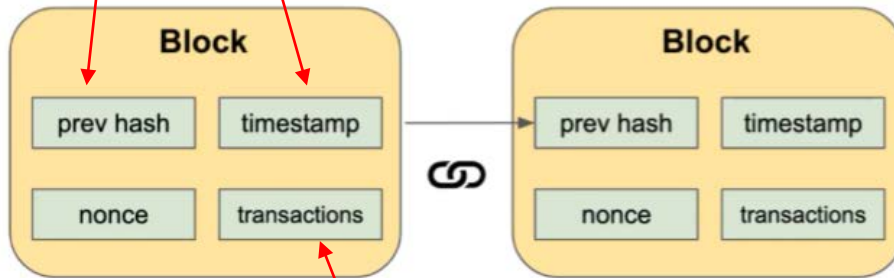
@bp.route('/', methods=['GET'])
def home():
    '''Mining 메인 화면'''
    # sync neighbor nodes
    return 'hello world'

@bp.route('/get_chain/', methods=['GET'])
def get_chain():
    '''블록체인 가져오기'''
    block_chain = get_blockchain()
    response = {
        # 'chain': 'under testing',
        'chain': block_chain.get('chain'),
    }
    return jsonify(response), 200
```

# 구현해야 하는 상황

이전 블록 해시 값

생성 시간



거래 내역



Pool

- Receiver 블록체인 주소: a23x644fce038
- Sender 블록체인 주소: 947e293acf467
- 금액: 1.0

- Receiver 블록체인 주소: a23x644fce038
- Sender 블록체인 주소: 947e293acf467
- 금액: 2.0



- Pool에 거래내역(transaction)들을 포함
- 마이닝을 통해 nonce 구하기
- 새로운 블록을 만들면 기존의 Pool은 초기화 되고 기존 거래내역은 다음 블록으로 전달  
(다음 슬라이드 참조)

# 데이터베이스 모델 구축 (models.py)

## ■ models.py

```
from mining import db

class Block(db.Model):
    '''블록 -> 단위 블록'''
    id = db.Column(db.Integer, primary_key=True)
    prev_hash = db.Column(db.String(300), nullable=True)
    nonce = db.Column(db.Integer, nullable=True)
    timestamp = db.Column(db.Float, nullable=True)

class Transaction(db.Model):
    '''거래정보: 하나의 블록에 담겨져 있는 거래 정보'''
    id = db.Column(db.Integer, primary_key=True)
    block_id = db.Column(db.Integer, db.ForeignKey('block.id'))
    send_addr = db.Column(db.String(300))
    recv_addr = db.Column(db.String(300))
    amount = db.Column(db.Float())
    blockchain = db.relationship(
        'Block',
        backref=db.backref('transactions')
    )
```

# 데이터베이스 생성

## ■ SQLite 데이터베이스 생성

- (venv)\$ export FLASK\_APP=mining # 윈도우일 경우 export 대신 set 명령어 사용
- (venv)\$ flask db init
- (venv)\$ flask db migrate
- (venv)\$ flask db upgrade