



로컬에서 빌드하기

목차

사전 설정

1. MySQL
2. Docker
2. Ganache 설정
3. Remix를 이용한 Contract 배포
4. [Frontend] `.env` 파일
3. [Backend] `/main/resources` 내 `application.properties`

사전 설정

1. MySQL

데이터베이스 생성

- charset: utf8
- collation: utf8_general_ci

2. Docker

```
docker run --name sgbg-redis -p 6379:6379 redis
```

- token 저장을 위한 redis 컨테이너

2. Ganache 설정

두 가지 방법 중 한 가지 방법 선택

2.1 ganache client 사용하기

Ganache

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK
0

GAS PRICE
20000000000

GAS LIMIT
6721975

HARDFORK
MUIRGLACIER

NETWORK ID
5777

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

WORKSPACE
QUICKSTART

SAVE

SWITCH

MNEMONIC

flower sleep worth vehicle syrup excite market chase nasty strike license law

HD PATH

m/44'/60'/0'/0/account_index

ADDRESS 0x97b2E73f146799eeA845b05F3902D9C648c96143	BALANCE 100.00 ETH	TX COUNT 0	INDEX 0	
ADDRESS 0xf93b7aFb12D2a8cf252602f0e26ac1A42Bb1a0B7	BALANCE 100.00 ETH	TX COUNT 0	INDEX 1	
ADDRESS 0xD2D146E70e4ff10043936C5624a6636395F0EbC6	BALANCE 100.00 ETH	TX COUNT 0	INDEX 2	
ADDRESS 0x7B47998839000a22E80646f1793602376F0452a9	BALANCE 100.00 ETH	TX COUNT 0	INDEX 3	
ADDRESS 0x8897f7bD2f1d6216fcAf34fFA690e265BBfbABbf	BALANCE 100.00 ETH	TX COUNT 0	INDEX 4	
ADDRESS 0x0C152e2756c321784Cc38c9Fd52A136824aDe001	BALANCE 100.00 ETH	TX COUNT 0	INDEX 5	
ADDRESS 0x11F6820B82EB765f9845c52fF1aF97d2f5ed972e	BALANCE 100.00 ETH	TX COUNT 0	INDEX 6	

- [주의] 7545번 포트로 띄워짐
 - application.properties에 기재하는 포트 번호 7545로 기재

2.2 (SGBG Default) ganache-cli 사용하기

```
ganache-cli -d -m -p 8545 -a 5
```

```
ganache-cli x + v
> ganache-cli -d -m -p 8545 -a 5
Ganache CLI v6.12.2 (ganache-core: 2.13.2)

Available Accounts
=====
(0) 0x90F8bf6A479f320ead074411a4B0e7944Ea8c9C1 (100 ETH)
(1) 0xFFcf8FDEE72ac11b5c542428B35EEF5769C409f0 (100 ETH)
(2) 0x22d491Bde2303f2f43325b2108D26f1eAbA1e32b (100 ETH)
(3) 0xE11BA2b4D45Eaed5996Cd0823791E0C93114882d (100 ETH)
(4) 0xd03ea8624C8C5987235048901fB614fDcA89b117 (100 ETH)

Private Keys
=====
(0) 0x4f3edf983ac636a65a842ce7c78d9aa706d3b113bce9c46f30d7d21715b23b1d
(1) 0x6cbcd15c793ce57650b9877cf6fa156fbef513c4e6134f022a85b1ffdd59b2a1
(2) 0x6370fd033278c143179d81c5526140625662b8daa446c22ee2d73db3707e620c
(3) 0x646f1ce2fdad0e6deeb5c7e8e5543bdde65e86029e2fd9fc169899c440a7913
(4) 0xadd53f9a7e588d003326d1cbf9e4a43c061aadd9bc938c843a79e7b4fd2ad743

HD Wallet
=====
Mnemonic:      myth like bonus scare over problem client lizard pioneer submit female collect
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Price
=====
2000000000

Gas Limit
=====
6721975

Call Gas Limit
=====
9007199254740991

Listening on 127.0.0.1:8545
```

- 8545번 포트로 띄우기

3. Remix를 이용한 Contract 배포

smart_contract/cash.sol 배포 후,

admin address와 private key 저장해두기

4. [Frontend] .env 파일

```
REACT_APP_KAKAO_KEY={KAKAO_KEY}
REACT_APP_KAKAO_API_KEY={KAKAO_API_KEY}
```

- KAKAO_KEY: 카카오 로그인 CLIENT KEY
- KAKAO_API_KEY: 카카오 맵 API KEY

3. [Backend] /main/resources 내 application.properties

```
server.port=8080

# BANNER
banner.location=classpath:banner.txt

# Charset of HTTP requests and responses. Added to the "Content-Type" header if not set explicitly.
server.servlet.encoding.charset=UTF-8
# Enable http encoding support.
server.servlet.encoding.enabled=true
# Force the encoding to the configured charset on HTTP requests and responses.
server.servlet.encoding.force=true

spring.devtools.livereload.enabled=true

# =====
# ~ DATA SOURCE
# =====
spring.jpa.hibernate.naming.implicit-strategy=org.springframework.boot.orm.jpa.hibernate.SpringImplicitNamingStrategy
spring.jpa.hibernate.naming.physical-strategy=org.springframework.boot.orm.jpa.hibernate.SpringPhysicalNamingStrategy
spring.jpa.hibernate.ddl-auto=update
spring.jpa.generate-ddl=true
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.show_sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect
spring.data.web.pageable.one-indexed-parameters=true
```

```

spring.datasource.url=jdbc:mysql://127.0.0.1:3306/bcssafy?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehavior=convertToNull&rewriteBatch
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.hikari.username=root
spring.datasource.hikari.password=12345

# =====
# = LOGGING
# =====
logging.file.name=./sgbg-log.log
logging.level.root=INFO
logging.level.org.springframework.web=ERROR
logging.level.org.springframework.boot=DEBUG
logging.level.org.apache.tiles=INFO
logging.level.com.sgbg=DEBUG

# Logging pattern for the console
logging.pattern.console=%d{HH:mm:ss} %c{r(%5p)} [%c] %m%n

# Logging pattern for file
logging.pattern.file=%d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level %logger{36} - %msg%n

#logging.file = logs/backend.log

# =====
# = Ethereum Client
# =====
spring.web3j.client.address=http://127.0.0.1:8545
spring.web3j.admin-client=true

# =====
# = Ethereum CA & EOA
# =====
#eth.encrypted.password=Pn001SN0SdhjNK5X2EHUPQ==
#eth.erc20.contract=0xcDcedAcea55DB472060042C6440960F65B9d9849
#eth.purchase.record.contract=0x0c2BC3cAB0036D1E51734B4D6e6cb8BCB046A2C7
#eth.admin.address=718ca8088ae9c120551defcbbecfedc3b9761a1f
#eth.admin.wallet.filename=admin.wallet

# Kakao REST API
kakao.client.id=b833a7474795b71dfb16b78fbcac80be
kakao.client.secret=ciUaftxmuIwrIwhekVk5c0mtH9H4KsNc
kakao.redirect.uri=http://localhost:3000/login

# Spring doc
springdoc.swagger-ui.path=/index.html
springdoc.swagger-ui.groups-order=desc
springdoc.swagger-ui.operations-sorter=method
springdoc.swagger-ui.disable-swagger-default-url=true
springdoc.swagger-ui.display-request-duration=true
springdoc.api-docs.path=/api-docs
springdoc.default-consumes-media-type=application/json
springdoc.default-produces-media-type=application/json
springdoc.paths-to-match=/*

# Redis for Token
spring.redis.host=localhost
spring.redis.port=6379

# Admin Address
eth.cash.contract={배포한_cash.sol_contract_address}
eth.admin.address={cash.sol에서_설정된_admin_address}
eth.admin.private={cash.sol에서_설정된_admin_private_key}

```