

포팅메뉴얼



개발 환경

1.1 운영체제

- Windows10

1.2 IDE

- IntelliJ IDEA 2023.2.2
- Visual Studio Code 1.83.0

1.3 Fron-End

- Typescript
- Vite.js
- Recoil
- Axios
- React

1.4 Back-End

- Javac flag: -parameters
- Java 17
- Spring Boot 버전 쓰기
 - Spring Web 3.1.3
 - Lombok
 - MySQL Driver
 - Spring Data JPA
- Gradle
- Spring Cloud Gateway
- FAST API
- Python 3.7.13
- anaconda win32

1.5 UI/UX

- Figma

1.6 DataBase

- MySQL
- Redis
- MongoDB

1.7 프로젝트 관리

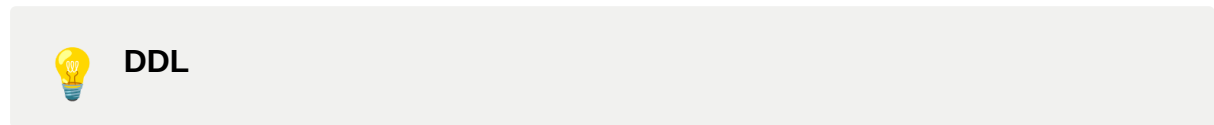
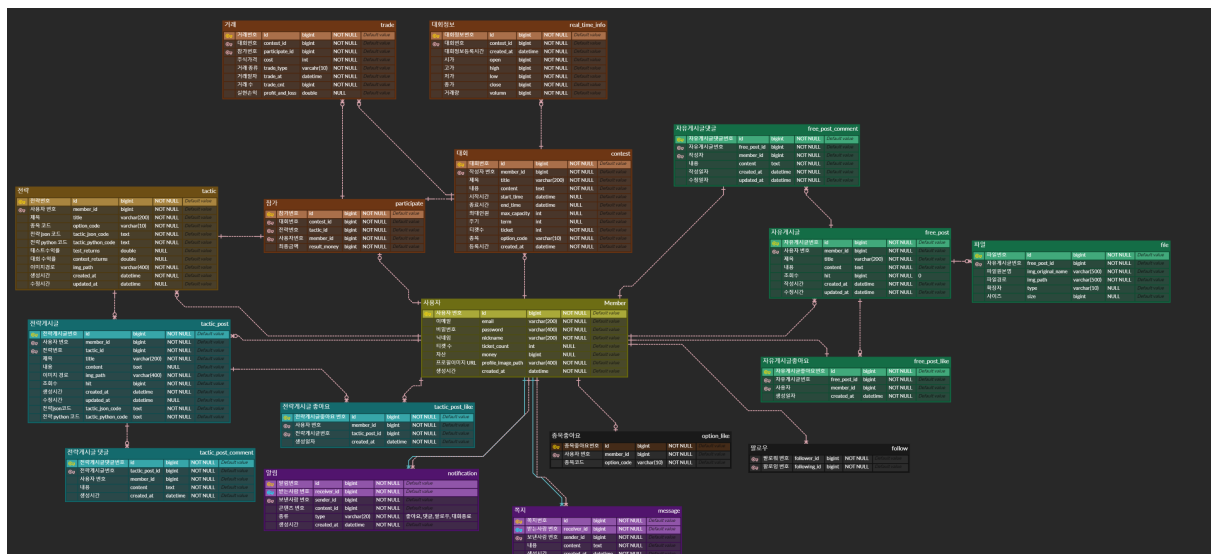
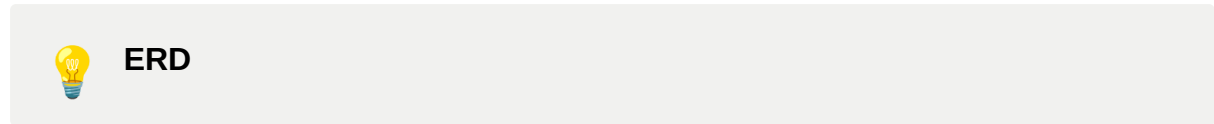
- GitLab
- Jira
- Notion

1.8프로젝트 관리1.8 커뮤니케이션

- Jenkins
- Docker
- Docker-compose
- Notion
- Mattermost
- Discord

1.8 커뮤니케이션

- Notion
- Mattermost
- Discord



- Tactic Server

```
create table contest
(
    id          bigint auto_increment
        primary key,
    member_id   bigint      not null,
    title       varchar(200) not null,
    content     text        not null,
    start_time  datetime    not null,
```

```

        end_time      datetime      not null,
        max_capacity  int            not null,
        term          int            not null,
        ticket        int            not null,
        option_code   varchar(10)    not null,
        created_at    datetime       null
    );

create table contest_real_time
(
    id          bigint auto_increment
        primary key,
    contest_id  bigint      null,
    created_at  datetime    null,
    open       bigint      null,
    high       bigint      null,
    low        bigint      null,
    close      bigint      null,
    vol        bigint      null,
    constraint contest_real_time_contest_id_fk
        foreign key (contest_id) references contest (id)
);

create table `option`
(
    option_code varchar(10) not null
        primary key,
    option_name varchar(255) null
);

create table option_like
(
    id          int auto_increment
        primary key,
    member_id   int            not null,
    option_code varchar(10) not null,
    constraint option_like_ibfk_1
        foreign key (option_code) references `option` (option_code)
);

create index option_code
    on option_like (option_code);

create table tactic
(
    id          bigint auto_increment
        primary key,
    member_id   bigint      not null,
    title       varchar(200) not null,
    option_code varchar(10)  not null,
    tactic_json_code text     not null,
    tactic_python_code text   not null,
    test_returns double      null,
    contest_returns double    null,
    img_path    varchar(400) not null,
    created_at  datetime     not null,

```

```

        updated_at      datetime      null
    );

create table participate
(
    id          bigint auto_increment
        primary key,
    contest_id  bigint not null,
    tactic_id   bigint null,
    member_id   bigint not null,
    result_money float not null,
    constraint contest_id
        foreign key (contest_id) references contest (id)
        on delete cascade,
    constraint tactic_id
        foreign key (tactic_id) references tactic (id)
);

create table trade
(
    id          bigint auto_increment
        primary key,
    contest_id  bigint      null,
    participate_id bigint not null,
    cost        int         null,
    trade_type  varchar(10) null,
    trade_at    datetime    null,
    trade_cnt   bigint      null,
    profit_and_loss double   null,
    trade_cost  int         not null,
    constraint participate_id
        foreign key (participate_id) references participate (id),
    constraint trade_contest_id_fk
        foreign key (contest_id) references contest (id)
);

```

- Board Server

```

create table free_post
(
    created_at datetime(6) null,
    hit        bigint      not null,
    id         bigint auto_increment
        primary key,
    member_id  bigint      not null,
    updated_at datetime(6) null,
    content    varchar(255) null,
    title      varchar(255) null
);

create table file

```

```

(
    free_post_id    bigint        null,
    id              bigint auto_increment
        primary key,
    size            bigint        not null,
    img_original_name varchar(255) null,
    img_path        varchar(255) null,
    type            varchar(255) null,
    constraint FKrt7978yqqjna4k3c3ntads6w4
        foreign key (free_post_id) references free_post (id)
);

create table free_post_comment
(
    created_at    datetime(6) null,
    free_post_id  bigint        null,
    id            bigint auto_increment
        primary key,
    member_id     bigint        not null,
    updated_at    datetime(6) null,
    content       varchar(255) null,
    constraint FKa037bkiv43y0hvde16ys6ctla
        foreign key (free_post_id) references free_post (id)
);

create table free_post_like
(
    created_at    datetime(6) null,
    free_post_id  bigint        null,
    id            bigint auto_increment
        primary key,
    member_id     bigint        not null,
    updated_at    datetime(6) null,
    constraint FKnrhkJn12ijgpwwkht21vypq1
        foreign key (free_post_id) references free_post (id)
);

create table tactic_post
(
    created_at    datetime(6) null,
    hit           bigint default 0 null,
    id            bigint auto_increment
        primary key,
    member_id     bigint        null,
    tactic_id     bigint        null,
    updated_at    datetime(6) null,
    content       varchar(255) null,
    img_path      varchar(255) null,
    tactic_json_code varchar(255) null,
    tactic_python_code varchar(255) null,
    title         varchar(255) null
);

create table tactic_post_comment
(
    created_at    datetime(6) null,

```

```

    id          bigint auto_increment
    primary key,
    member_id    bigint          null,
    tactic_post_id bigint        null,
    updated_at   datetime(6)    null,
    content      varchar(255)    null,
    constraint FKcojaba2sxlmcu2eg3dqy8mrod
        foreign key (tactic_post_id) references tactic_post (id)
);

create table tactic_post_like
(
    created_at    datetime(6) null,
    id            bigint auto_increment
    primary key,
    member_id     bigint       null,
    tactic_post_id bigint       null,
    updated_at    datetime(6) null,
    constraint FK8tylcjao14095mrburke5of1b
        foreign key (tactic_post_id) references tactic_post (id)
);

```



프로젝트 내 사용 외부 서비스

- Blockly

Blockly 브랜드 표현 | Google for Developers

Blockly 속성을 올바르게 지정하는 방법

 <https://developers.google.com/blockly/guides/app-integration/attribution?hl=ko>

Google for Developers

- 대신증권

CYBOS 5	CYBOS i	CYBOS Onedlick	CYBOS Trader
---------	---------	----------------	--------------

CYBOS 5

대신이 만들고 세계가 사용하는 대표 HTS프로그램

한층 강화시킨 CYBOS 5가 더욱 강력한 시스템으로 탄생하여 신개념 사이버 거래 시스템으로 고객 여러분과 한발 앞선 사이버 투자 세상의 동반자가 될 것입니다.



[다운로드](#) [도움말](#) [USB마법사](#)

CYBOS 5 > 온라인트레이딩 - 대신증권

57초 후 자동 접속 종료 예정입니다!

https://money2.daishin.com/E5/WTS/Customer/GuideTrading/DW_Cybos5.aspx?p=4545&v=3383&m=1098

• 한국투자증권

KIS Developers

<https://apiportal.koreainvestment.com/intro>



시연 시나리오

4.1 회원가입 및 로그인

- 회원가입 버튼 클릭
- 이메일 인증
- 회원가입 성공
- 로그인
- 로그인 성공

4.2 메인페이지

4.3 전략 생성 및 테스트

- 종류 별 블록들 클릭, 소개
- 가이드 블록 소개
- 가이드 블록 수정
- 테스트하기
- 결과 화면 이동
- 매매 내역 및 매매내역 상세 소개
- 저장하기

4.4 전략 게시판

- 전략 게시글 작성
- 전략 선택
- 전략 게시글 조회
- 다른 사람 전략 복사

4.5 자유 게시판

- 자유게시글 조회

4.6 마이페이지

- 어워드 소개
- 자산 충전
- 티켓 충전
- 쪽지
- 쪽지 삭제
- 나의 게시글
- 기록 관리
- 전략 목록

4.7 대회

- 대회 참가 신청
- 참가할 전략 선택
- 대회 참가 신청
- 현재 진행 중 대회 소개
- 실시간 데이터 확인
- 종료된 대회 소개
- 종료된 대회 결과 확인