Tony의 주간 발표~

목차로는

- Git floW
- Test cODe
- eTc

Git flow

Git flow

Git flow

Git flow

Git flow

Git flow

Git flow

‘git flow’는 깃에서 제공하는 강력한 브랜칭 기능을 활용한 변경이력 관리 전략이다.

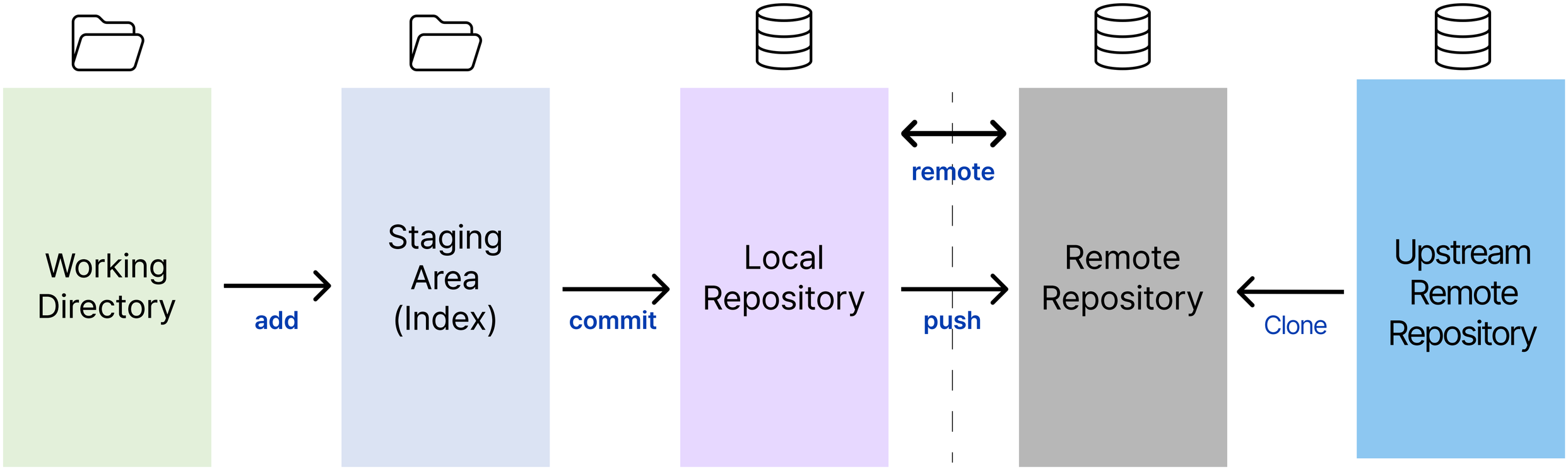
이 전략은 상황에 따라서 다양한 변화가 가능하다.

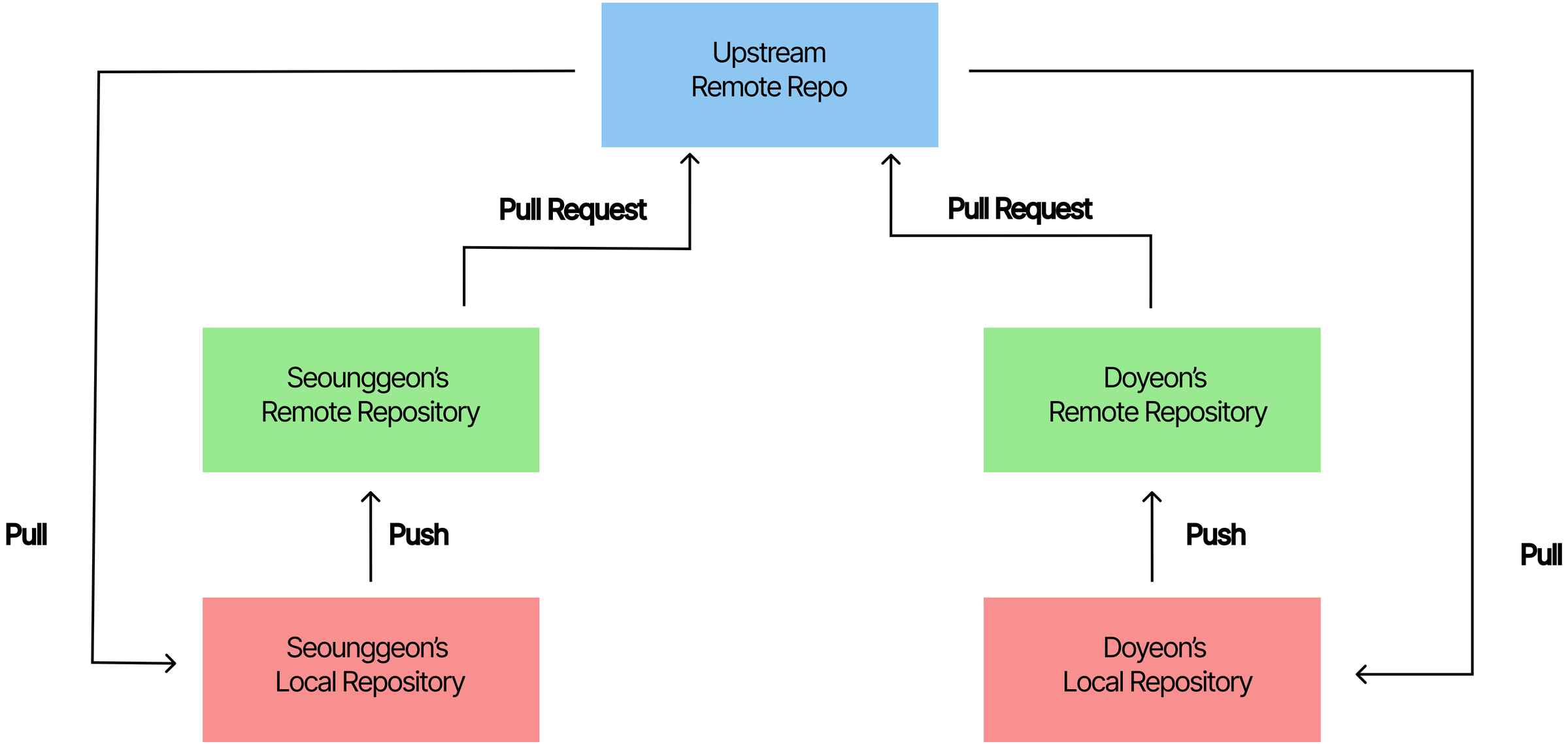
Git flow

깃플로우는 다양한 변형된 형태의 전략을 세울수 있으며
깃헙github, 깃랩gitlab에서 제공하는 방식이 있다.

CoNoE Git flow

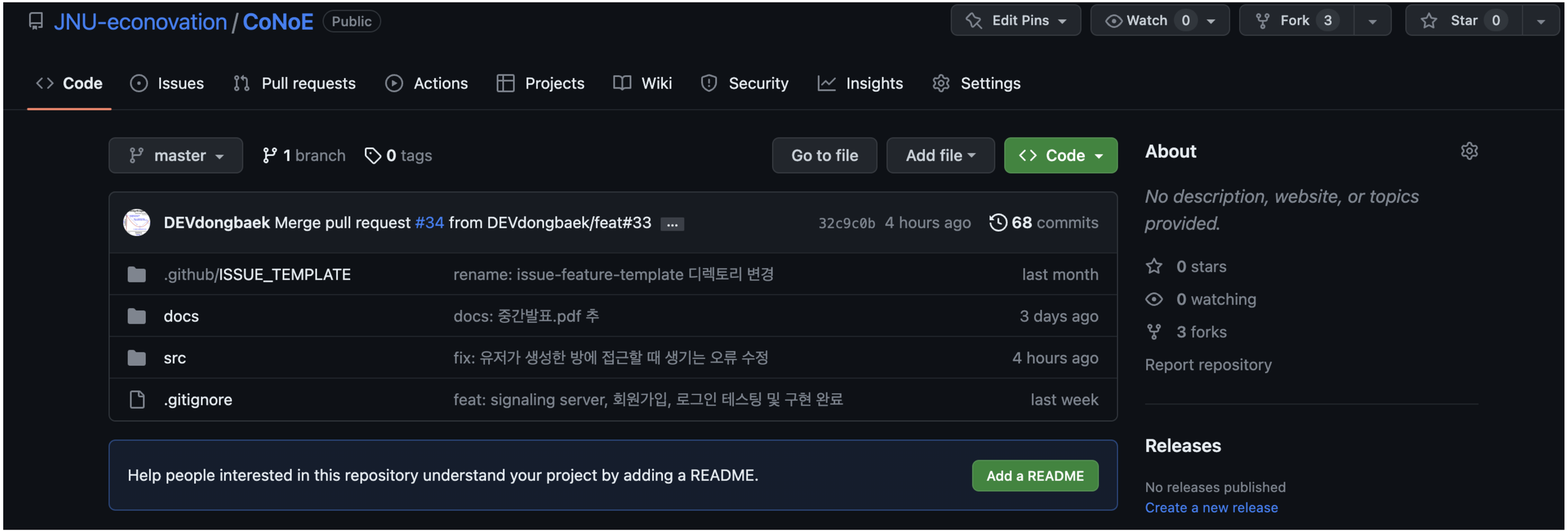
02 Git의 명령어를 알아보자!




Upstream
Remote Repo
Pull Request
Pull Request
Seounggeon's
Remote Repository
Doyeon's
Remote Repository
Pull
Push
Push
Pull
Seounggeon's
Local Repository
Doyeon's
Local Repository

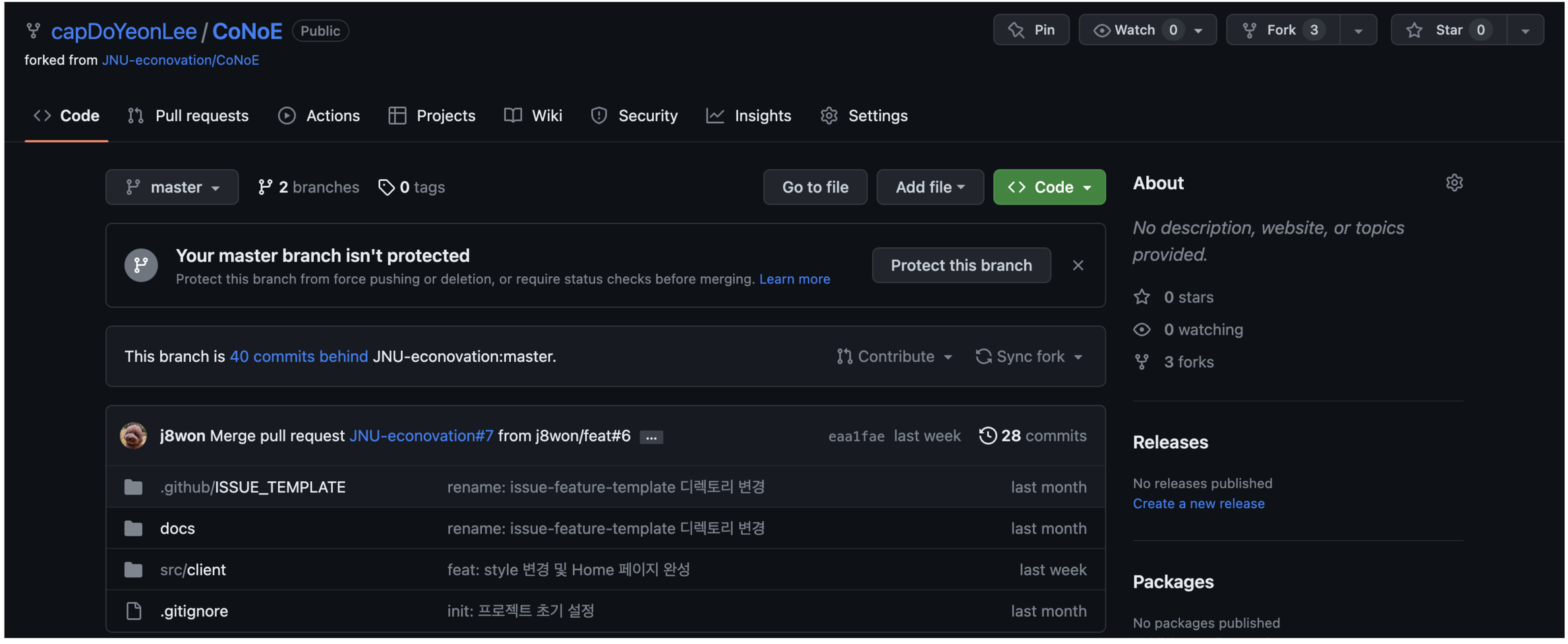
Upstream
Remote Repo

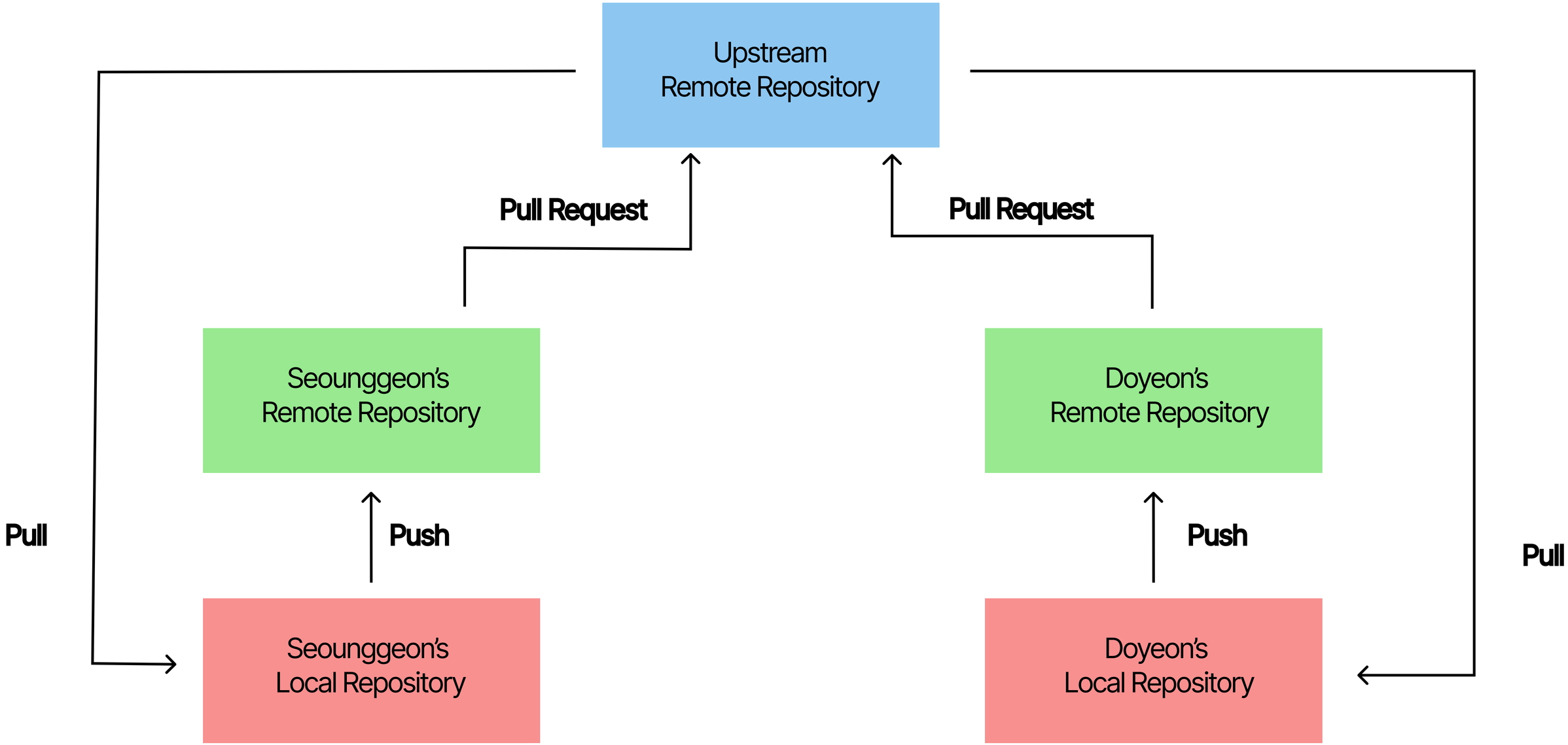
Upstream
Remote Repository



Doyeon's
Remote Repository

Remote Repository

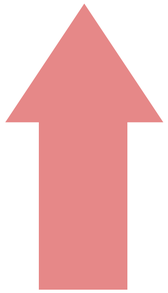



Upstream
Remote Repository
Pull Request
Pull Request
Seounggeon's
Remote Repository
Doyeon's
Remote Repository
Pull
Push
Push
Pull
Seounggeon's
Local Repository
Doyeon's
Local Repository

이런 워크플로우를 사용하는 이유가 뭘까요??

맞추면 커피 사드림 ㅋ

독립적인 공간(Origin Repository)



위험(충돌)없이 다양한 시도를 할 수 있다!!!!

We can do anything!!!!

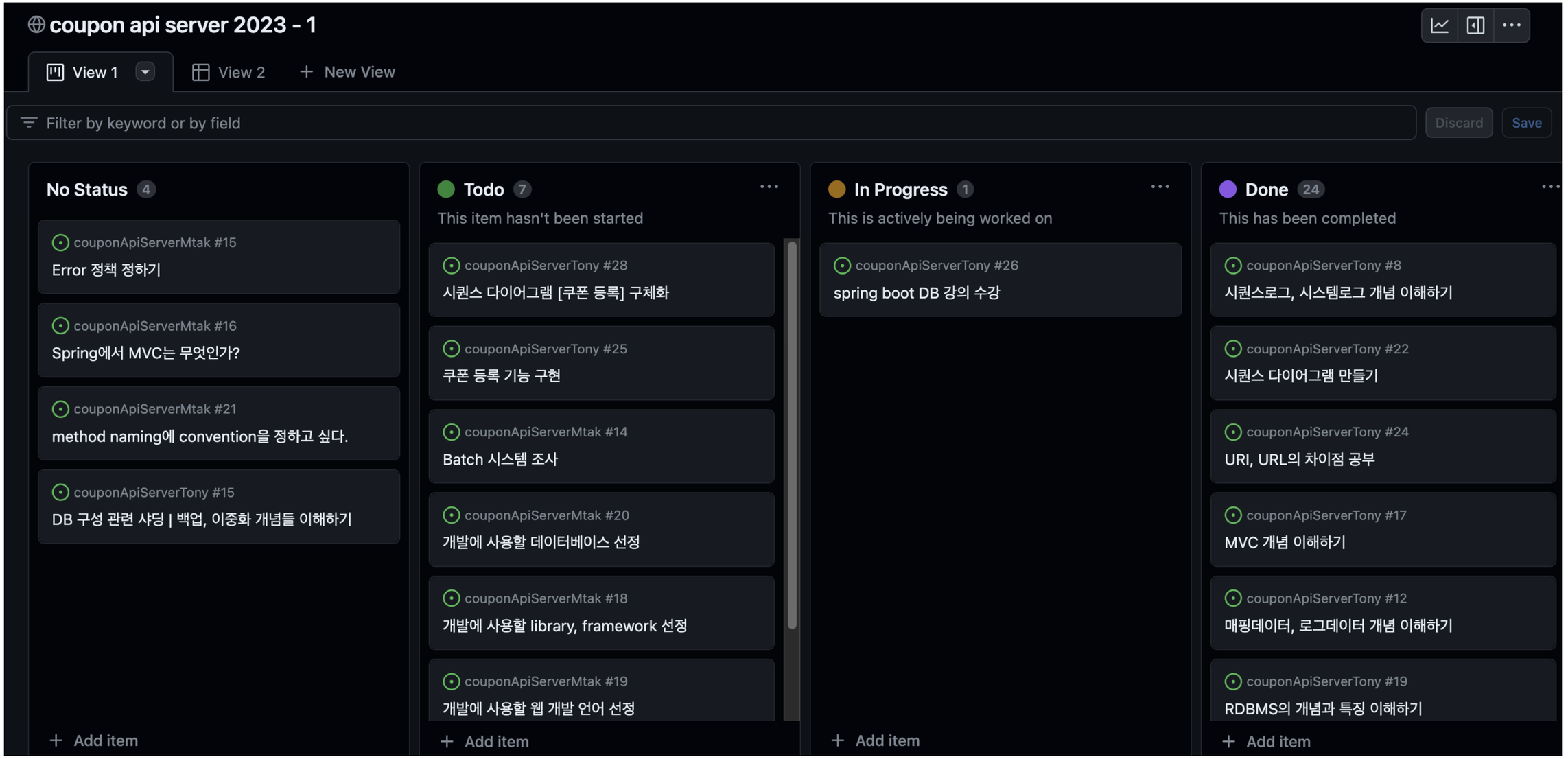
Upstream Repository의 무결성을 위함!!

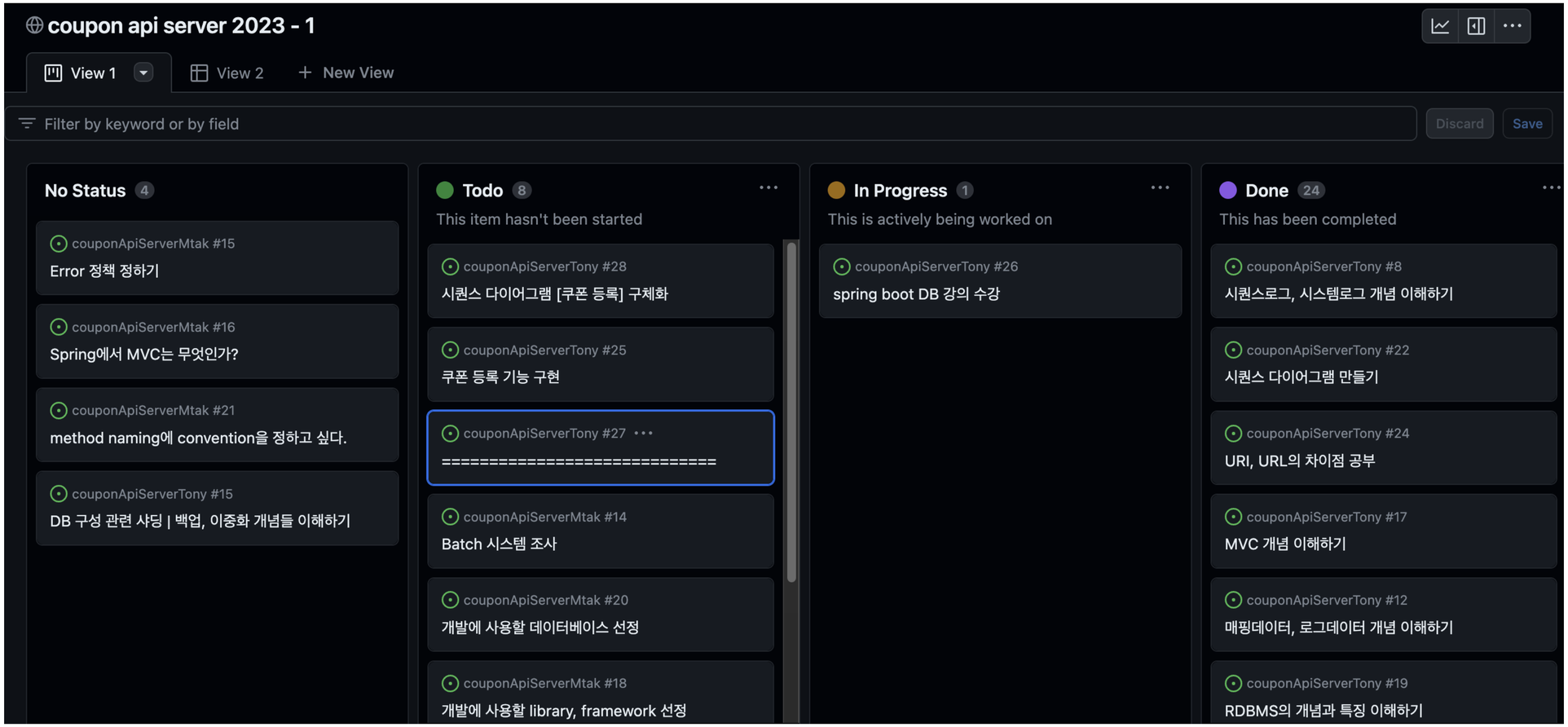
무결성이란???

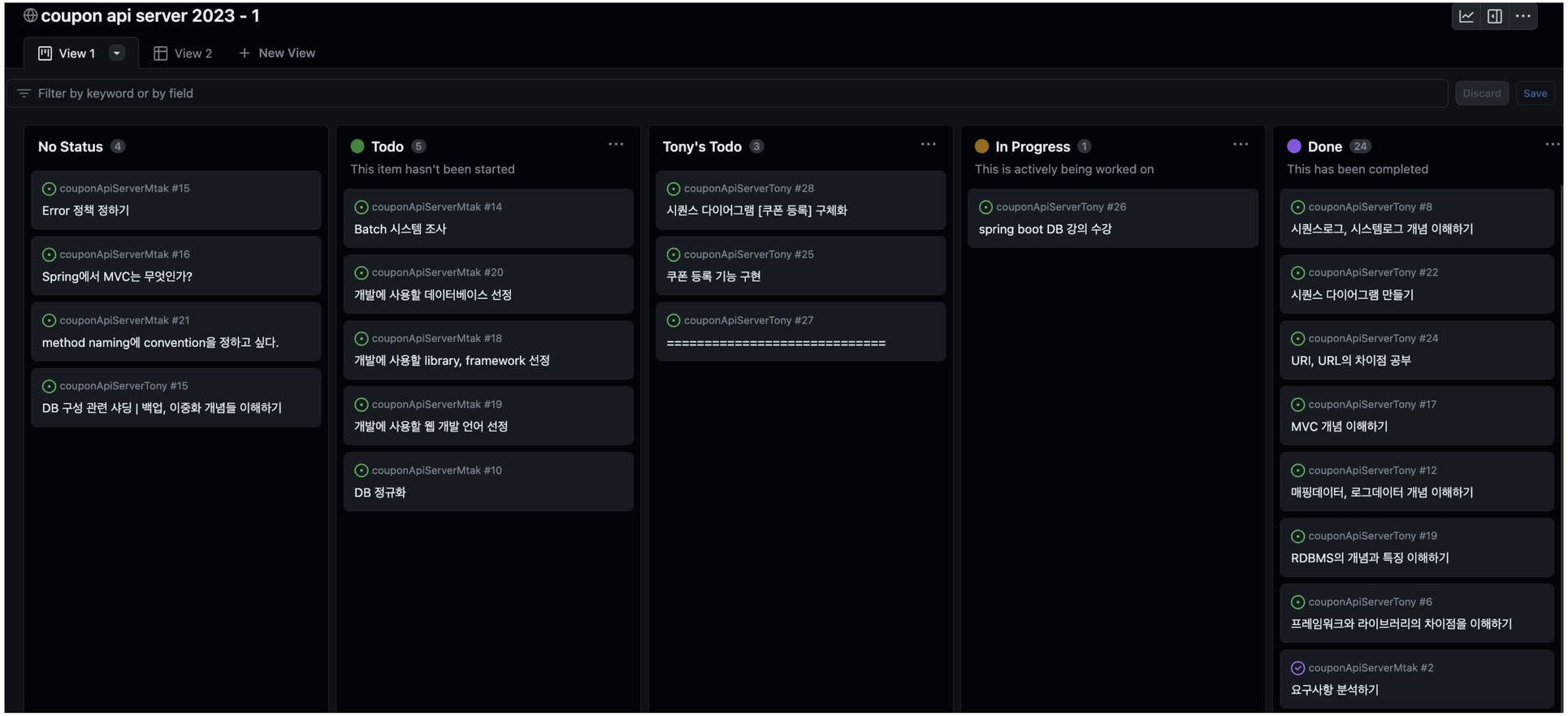
컴퓨팅 분야에서

유지 및 보증

SurplusTeam

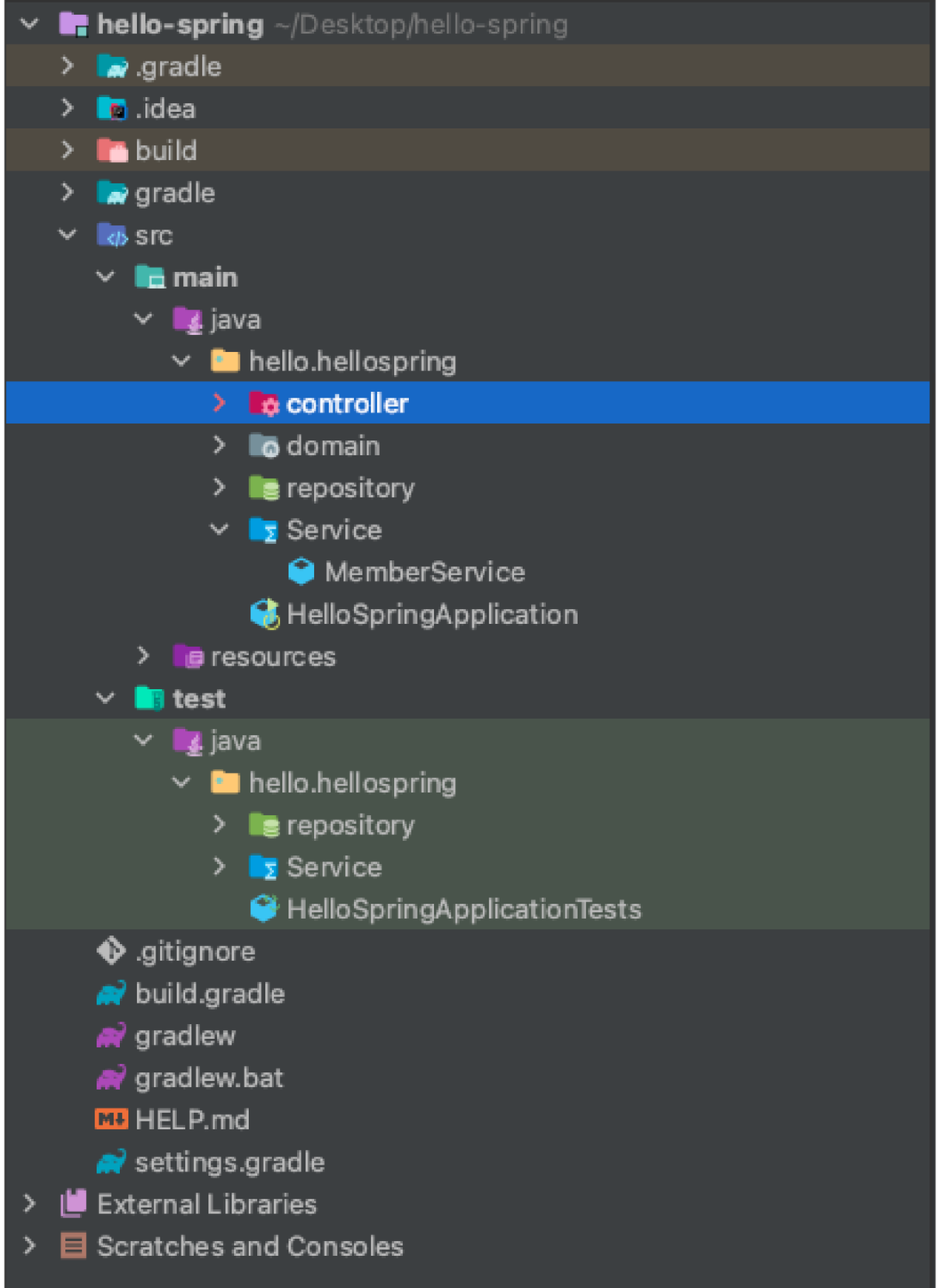

coupon api server 2023 - 1
View 1
View 2
New View
Filter by keyword or by field
Discard
Save
No Status 4
couponApiServerMtak #15
Error 정책 정하기
couponApiServerMtak #16
Spring에서 MVC는 무엇인가?
couponApiServerMtak #21
method naming에 convention을 정하고 싶다.
couponApiServerTony #15
DB 구성 관련 샤딩 | 백업, 이중화 개념들 이해하기
Add item
Todo 7
This item hasn't been started
couponApiServerTony #28
시퀀스 다이어그램 [쿠폰 등록] 구체화
couponApiServerTony #25
쿠폰 등록 기능 구현
couponApiServerMtak #14
Batch 시스템 조사
couponApiServerMtak #20
개발에 사용할 데이터베이스 선정
couponApiServerMtak #18
개발에 사용할 library, framework 선정
couponApiServerMtak #19
개발에 사용할 웹 개발 언어 선정
Add item
In Progress 1
This is actively being worked on
couponApiServerTony #26
spring boot DB 강의 수강
Add item
Done 24
This has been completed
couponApiServerTony #8
시퀀스로그, 시스템로그 개념 이해하기
couponApiServerTony #22
시퀀스 다이어그램 만들기
couponApiServerTony #24
URI, URL의 차이점 공부
couponApiServerTony #17
MVC 개념 이해하기
couponApiServerTony #12
매핑데이터, 로그데이터 개념 이해하기
couponApiServerTony #19
RDBMS의 개념과 특징 이해하기
Add item


coupon api server 2023 - 1
View 1
View 2
New View
Filter by keyword or by field
Discard
Save
No Status 4
couponApiServerMtak #15
Error 정책 정하기
couponApiServerMtak #16
Spring에서 MVC는 무엇인가?
couponApiServerMtak #21
method naming에 convention을 정하고 싶다.
couponApiServerTony #15
DB 구성 관련 샤딩 | 백업, 이중화 개념들 이해하기
Todo 8
This item hasn't been started
couponApiServerTony #28
시퀀스 다이어그램 [쿠폰 등록] 구체화
couponApiServerTony #25
쿠폰 등록 기능 구현
couponApiServerTony #27
==========
couponApiServerMtak #14
Batch 시스템 조사
couponApiServerMtak #20
개발에 사용할 데이터베이스 선정
couponApiServerMtak #18
개발에 사용할 library, framework 선정
In Progress 1
This is actively being worked on
couponApiServerTony #26
spring boot DB 강의 수강
Done 24
This has been completed
couponApiServerTony #8
시퀀스로그, 시스템로그 개념 이해하기
couponApiServerTony #22
시퀀스 다이어그램 만들기
couponApiServerTony #24
URI, URL의 차이점 공부
couponApiServerTony #17
MVC 개념 이해하기
couponApiServerTony #12
매핑데이터, 로그데이터 개념 이해하기
couponApiServerTony #19
RDBMS의 개념과 특징 이해하기


coupon api server 2023 - 1
View 1
View 2
New View
Filter by keyword or by field
Discard
Save
No Status 4
couponApiServerMtak #15
Error 정책 정하기
couponApiServerMtak #16
Spring에서 MVC는 무엇인가?
couponApiServerMtak #21
method naming에 convention을 정하고 싶다.
couponApiServerTony #15
DB 구성 관련 샤딩 | 백업, 이중화 개념들 이해하기
Todo 5
This item hasn't been started
couponApiServerMtak #14
Batch 시스템 조사
couponApiServerMtak #20
개발에 사용할 데이터베이스 선정
couponApiServerMtak #18
개발에 사용할 library, framework 선정
couponApiServerMtak #19
개발에 사용할 웹 개발 언어 선정
couponApiServerMtak #10
DB 정규화
Tony's Todo 3
couponApiServerTony #28
시퀀스 다이어그램 [쿠폰 등록] 구체화
couponApiServerTony #25
쿠폰 등록 기능 구현
couponApiServerTony #27
==========
In Progress 1
This is actively being worked on
couponApiServerTony #26
spring boot DB 강의 수강
Done 24
This has been completed
couponApiServerTony #8
시퀀스로그, 시스템로그 개념 이해하기
couponApiServerTony #22
시퀀스 다이어그램 만들기
couponApiServerTony #24
URI, URL의 차이점 공부
couponApiServerTony #17
MVC 개념 이해하기
couponApiServerTony #12
매핑데이터, 로그데이터 개념 이해하기
couponApiServerTony #19
RDBMS의 개념과 특징 이해하기
couponApiServerTony #6
프레임워크와 라이브러리의 차이점을 이해하기
couponApiServerMtak #2
요구사항 분석하기

Test
cODe

Test cODe


hello-spring ~/Desktop/hello-spring
.gradle
.idea
build
gradle
src
main
java
hello.hellospring
controller
domain
repository
Service
MemberService
HelloSpringApplication
resources
test
java
hello.hellospring
repository
Service
HelloSpringApplicationTests
.gitignore
build.gradle
gradlew
gradlew.bat
HELP.md
settings.gradle
External Libraries
Scratches and Consoles

Main

```java
ghtyu21@naver.com <ghtyu21@naver.com> *
public class MemoryMemberRepository implements MemberRepository{

    private static Map<Long, Member> store = new HashMap<>();
    //0뒤에 대문자 "L"을 붙인 이유 -> 0으로 초기화 했다면 Int로 인식, long 타입 값을 나타내기 위해서는 숫자 뒤에 반드시 L을 붙여야한다.
    private static long sequence = 0L;

    ghtyu21@naver.com <ghtyu21@naver.com>
    public void clearStore() { store.clear(); }

    ghtyu21@naver.com <ghtyu21@naver.com>
    @Override
    public Member save(Member member) {
        member.setId(++sequence);
        store.put(member.getId(), member);
        return member;
    }
    ghtyu21@naver.com <ghtyu21@naver.com>
    @Override
    public Optional<Member> findById(Long id) {
        // 자바 문법에서 <>가 잘 이해가 안간다. 괄호안에 타입으로
        return Optional.ofNullable(store.get(id));
        //...
    }
    ghtyu21@naver.com <ghtyu21@naver.com>
    @Override
    public Optional<Member> findByName(String name) {

        return store.values().stream()
                .filter(member -> member.getName().equals(name)) // getName이 name과 동일한지 확인
                .findAny();  //자바 8 문법 String
    }
    ghtyu21@naver.com <ghtyu21@naver.com>
    @Override
    public List<Member> findAll() {

        return new ArrayList<>(store.values()); // Map<Long, Member> 여기서 store.valuse()는 Member를 의미?
    }
}
```

teSt

```java
ghtyu21@naver.com <ghtyu21@naver.com> *
public class MemoryMemberRepositoryTest {

    MemoryMemberRepository repository = new MemoryMemberRepository();
    ghtyu21@naver.com <ghtyu21@naver.com>
    @AfterEach  // 공용 데이터 제거
    public void afterEach() { repository.clearStore(); }

    ghtyu21@naver.com <ghtyu21@naver.com> *
    @Test
    public void save(){
        Member member = new Member();
        member.setName("Tony");

        repository.save(member);

        // 여기 result의 타입을 왜 Member type으로? #TODO
        Member result = repository.findById(member.getId()).get();
        System.out.println(repository.findById(member.getId()).get());
        //result의 값은 :hello.hellospring.domain.Member@29a0cdb -> HashMap이라서 Hash값으로 매핑이 된건가??
//        System.out.println("result의 값은 :" + result);
//        System.out.println("result =" + (result == member));
        assertEquals(member, result); //기대하는 값, 현재 값
        assertThat(member).isEqualTo(result);
    }
    ghtyu21@naver.com <ghtyu21@naver.com>
    @Test
    public void findByName(){
        Member member1 = new Member();
        member1.setName("Tony");
        repository.save(member1);

        Member member2 = new Member(); // shift + f6
        member2.setName("Doyeon");
        repository.save(member2);

        Member result = repository.findByName("Tony").get();

        assertThat(result).isEqualTo(member1);
    }
```