



## 제품소프트웨어 패키징 (Git과 GitHub를 활용한 소스코드 관리)

# Issue 생성 및 관리



## 학습내용

- 프로젝트에서의 Issue 관리
- GitHub의 Issue 관리



## 학습목표

- 프로젝트 진행에 있어 Issue관리의 방법에 대해 설명할 수 있다.
- GitHub의 Issue기능을 실습하고 실제 적용해 볼 수 있다.



## 프로젝트에서의 Issue 관리

### 1. Issue 관리 방법

#### 1) 다양한 내용의 효율적인 관리

- Issue 생성
- 진행상황 관리
  - 단순히 버그를 접수 받고 관리자만 알고 있거나 따로 정리하지 않으면 문제점에 대해서 파악 불가능
  - 버그나 요구사항 등 다양한 정보를 등록하고, 그 이력을 추적하기 위해 다양한 Issue관리 시스템 사용

#### 2) Issue관리시스템 운영방침

- 모든 Issue에 대해 무조건 등록 - 지속적인 관리 추적 가능
- 고객 등의 참여를 유도하여 프로젝트 확인 및 수정사항에 대한 즉각적인 피드백
- 개발자가 직접 기간을 산정, PM이 향후 보완을 해 주는 방향으로 정리
- 고객에게도 Issue관리 시스템을 수시로 체크하도록 권장
- 관리자는 Issue를 일일 단위로 관리하여 그 수행 결과를 Issue관리 시스템에 반영
- 초기 개발 보다 지속적인 Issue관리를 통해 개선되는 것을 잘 관리해야 좋은 프로젝트로 진행 가능



## 프로젝트에서의 Issue 관리

### 2. Issue 관리 팁

- 오픈소스 프로젝트에 참여하는 컨트리뷰터가 많아야 함
- 오픈소스 컨트리뷰터 : 오픈소스에 직접적으로 참여하는 사람
  - 직접 오픈소스에서 단 한 줄이라도 소스코드를 만들어서 반영하는 경우
  - 소스코드를 같이 검토하거나 그에 대한 의견을 제시하는 경우
- 오픈소스도 많은 컨트리뷰터의 참여가 성패를 좌우하기 때문에 많은 사람들이 참여할 수 있는 방법을 제공
- 프로젝트 관리에도 많은 노력을 기울여야 함



## 프로젝트에서의 Issue 관리

### 2. Issue 관리 팁

#### 1) 좋은 라벨 사용

- 직관적이고 유연하게, 단순하게 구성하여 많은 개발자들의 참여 유도
- 관리가 되어있지 않다면, 수많은 Issue들에 대해서 자신과 관심 없는 것이 나열되어 있는 것을 보고 그냥 지나치게 됨
  - 특정 라벨 사용 : 관심 있는 부분에 대해서 어떤 행동을 할 지 찾아보고 참여
  - 많은 라벨 사용 : 분류하는 기준 모호 등 관리의 어려움만 증가

#### 2) 일의 종류 분류

- 현재 Issue에 대한 분류
  - 완전 새로운 일 VS 기존의 개선
- 컨트리뷰터들이 해당 Issue 내용을 읽기 전 파악 가능
- 개발 위주 Issue의 경우 카테고리 분류
  - 카테고리 라벨 없는 경우 : 모든 개발자들이 Issue를 보고 고쳐야 할 부분 선택
  - 카테고리 라벨 있는 경우 : 테스트/버그수정/리팩토링 등 개발자들이 적합한 Issue 선택 가능
- 자유롭게 여러 부분에서 수정되고 발전되어 가는 오픈소스 프로젝트
  - 일의 종류를 나누어 놓는다면 각 개발자들이 자신이 중요하다 생각하거나 관심 있는 부분에 더 많은 시간을 투자할 수 있음
- 라벨의 예
  - Refactoring : 실제 코딩 보다는 전체적인 구조를 보고 수정하는 것을 좋아하는 컨트리뷰터가 많이 활동
  - Bugfix : 해당 오류에 대해 가장 시급히 수정되어서 활용되기를 바라는 사람이 더 많이 참여하고 빠르게 수정



## 프로젝트에서의 Issue 관리

### 2. Issue 관리 팁

#### 3) 난이도 설정

- 각 Issue의 해결을 위한 난이도를 초급/중급/고급 등으로 나누어서 등록해 놓는다면 프로젝트 업무 지시에 있어 큰 도움
- 라벨의 예
  - Beginner : 신입사원은 Beginner로 라벨링 된 Issue 위주로 검색하여 업무진행
  - Bigger project : 고급 개발자는 고급 Issue를 확인하지 못하여 낭비되는 시간 감소
- 난이도 미관리 : 중요함 보다는 쉬운 기능 위주로 구현이 되는 것이 프로젝트가 진행이 잘 되고 있다고 판단
- 난이도 관리
  - 난이도와 그 진척도에 따라 전체적인 파악 가능
  - 개발자들도 자신이 도전할 수 있는 난이도에 더욱 더 많은 시간 할애

#### 4) 우선순위 라벨링

- 상용 프로젝트의 경우 비즈니스적으로 급한 문제들이 Issue로 등록
- 오픈소스 작업 단점
  - 사람들이 자신이 하고 싶은 일만 골라서 작업
  - 프로젝트 전체적인 흐름에서의 중요성은 크게 신경 쓰지 않음
- 개발자들에게 이것이 꼭 고쳐지기를 원한다면, Critical과 같은 라벨 설정 등록
- 프로젝트 진행을 위해서 라벨을 제대로만 설정해 놓아도 별도의 지시 등의 업무 없이 Issue를 좀더 쉽게 처리



## GitHub의 Issue 관리

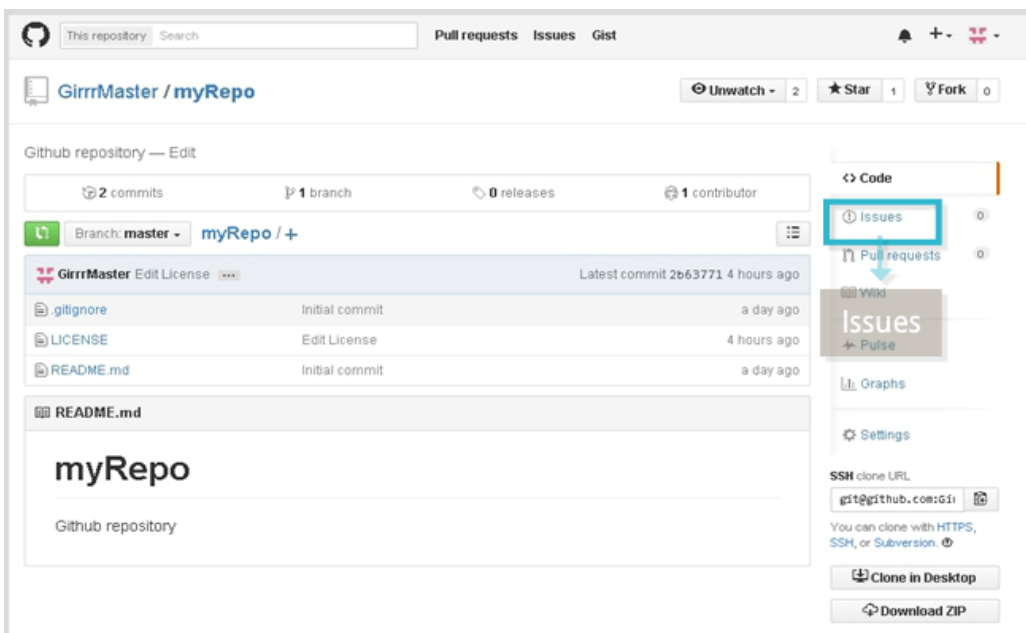
### 1. Issue 화면 구성

#### 1) Issue시스템 접속

- GitHub 웹페이지 접속



- Repository화면 좌측 Issues클릭



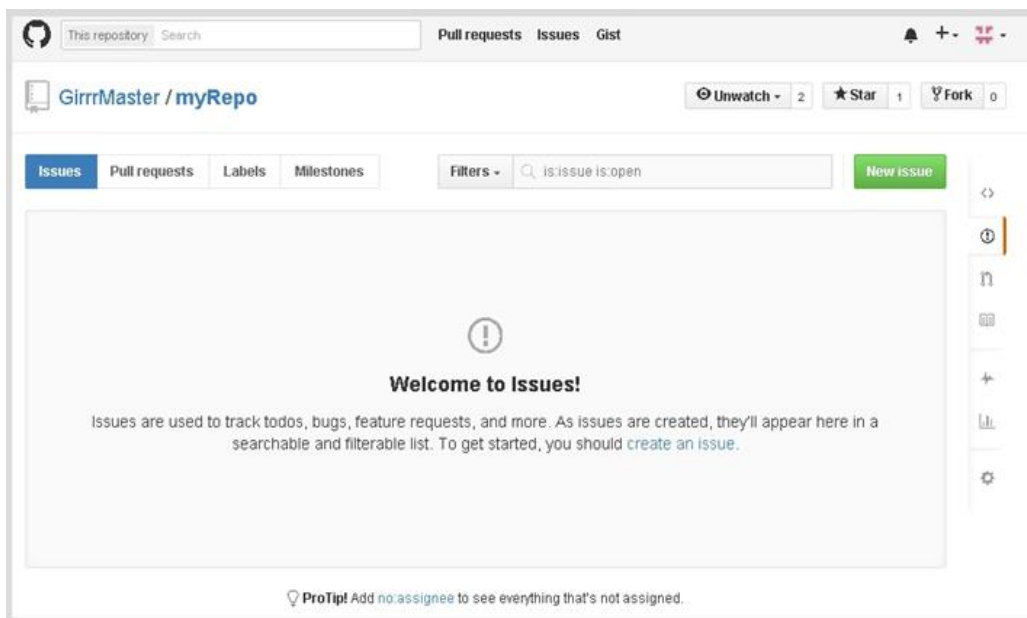


## GitHub의 Issue 관리

### 1. Issue 화면 구성

#### 1) Issue시스템 접속

- 생성된 Issues와 해결된 Issues 확인



- 수치를 통해 현재 프로젝트의 진척 상황을 파악하는 용도로 활용



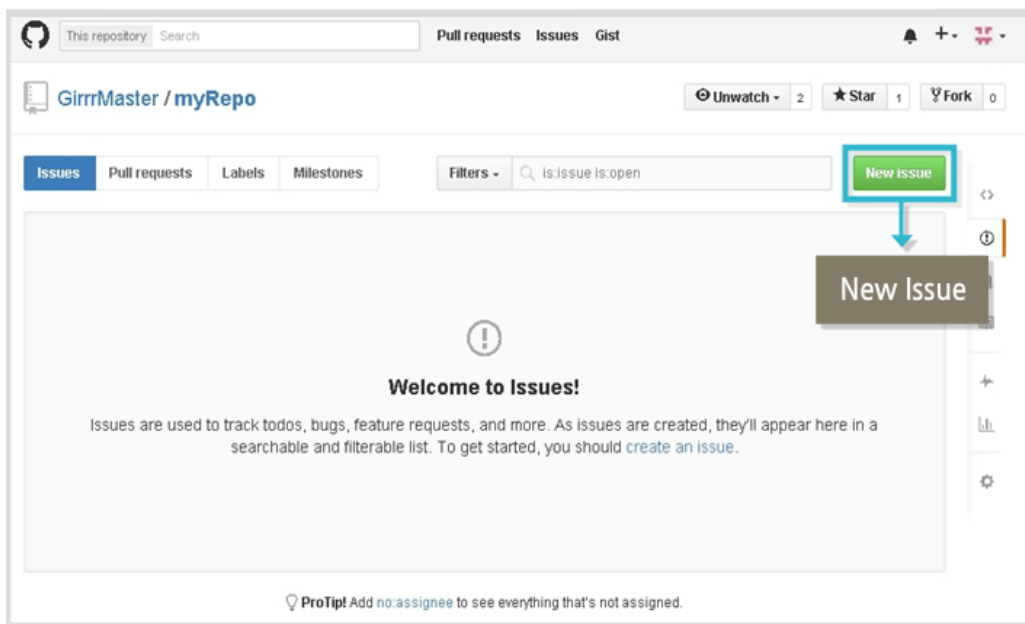


## GitHub의 Issue 관리

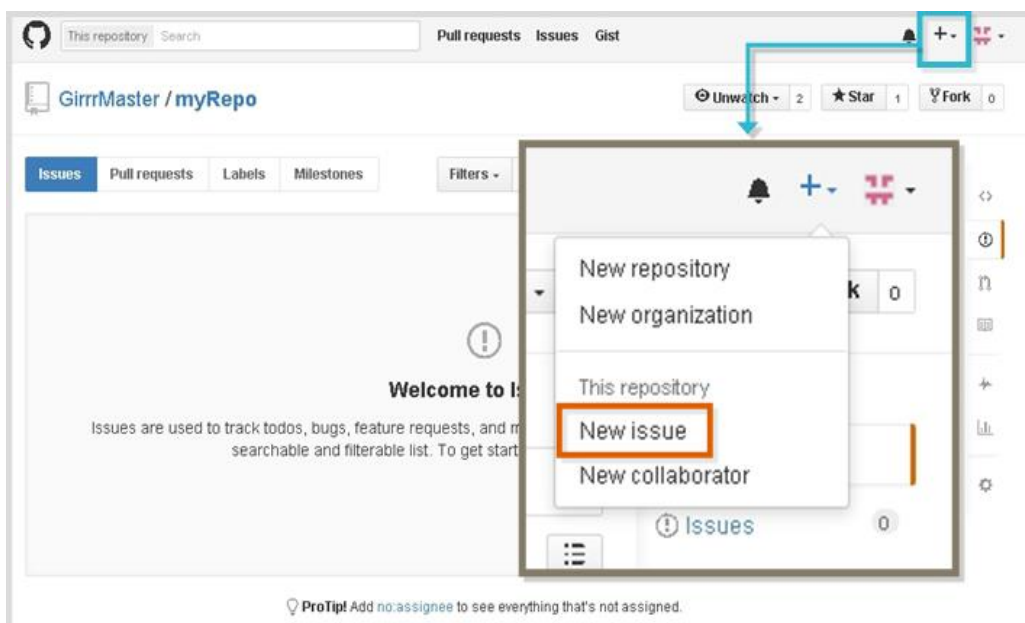
### 1. Issue 화면 구성

#### 2) 새로운 Issue 생성

- New issue 버튼 클릭



- 화면 상단 '+' 버튼 클릭



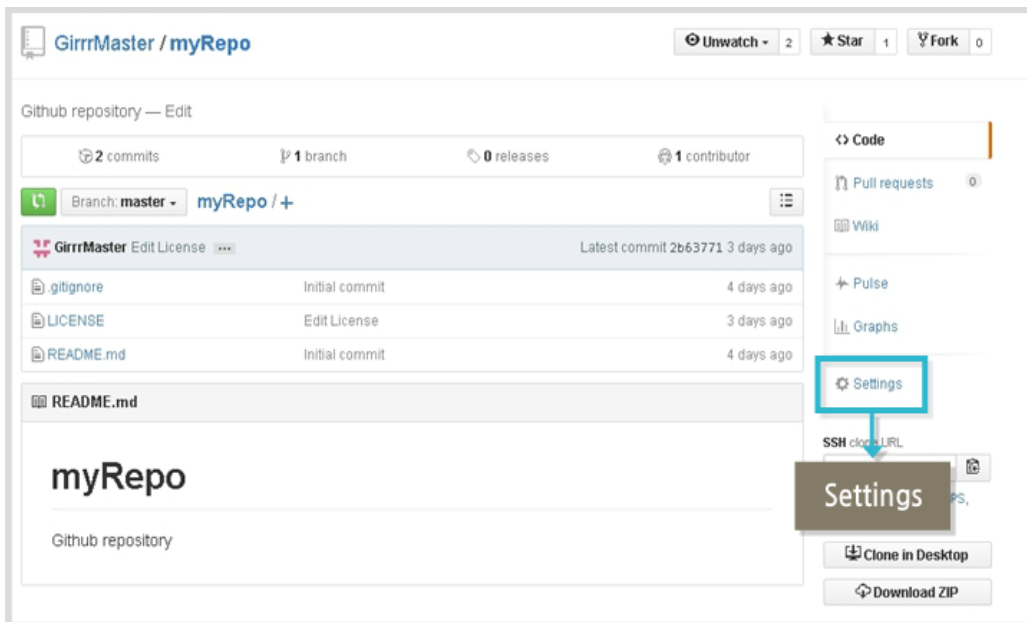


## GitHub의 Issue 관리

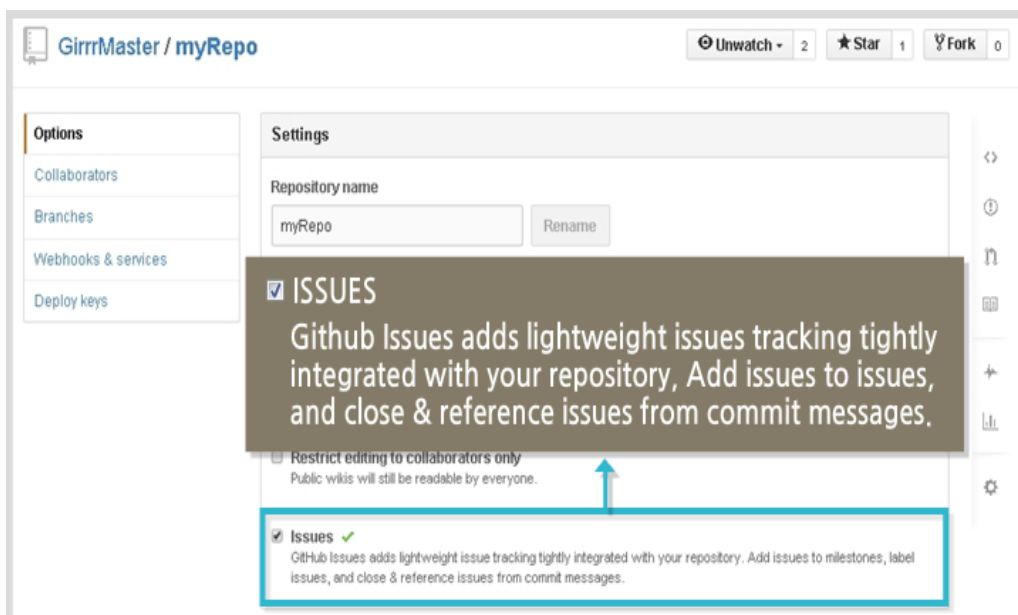
### 1. Issue 화면 구성

#### 3) Issues 메뉴 추가

- Settings 메뉴 클릭



- Issues 항목 클릭



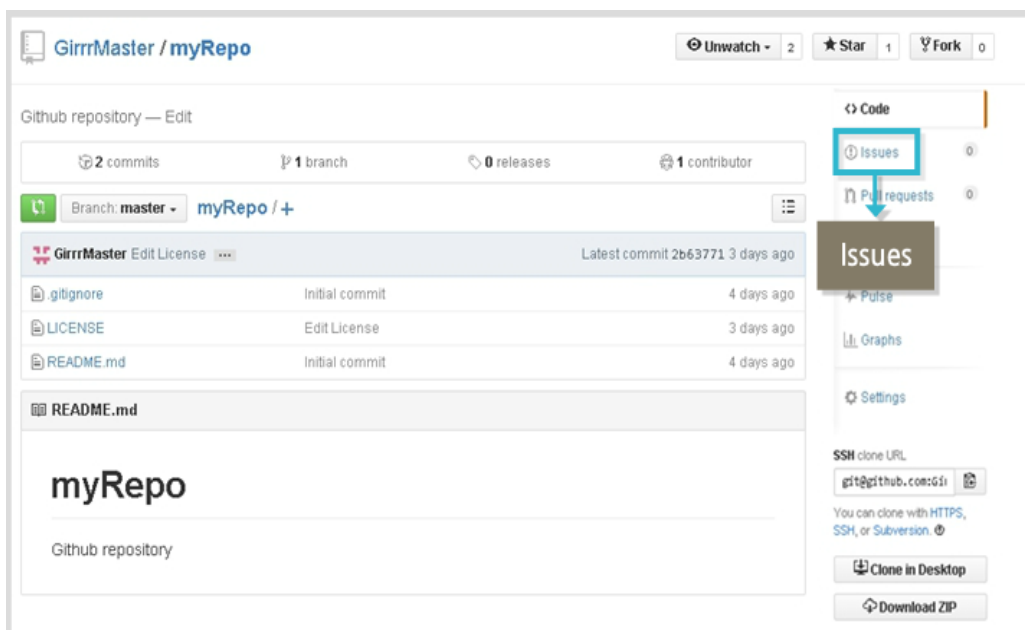


## GitHub의 Issue 관리

### 1. Issue 화면 구성

#### 3) Issues 메뉴 추가

- Issues메뉴 추가 확인





## GitHub의 Issue 관리

### 2. Issue 생성

#### 1) 새로운 Issue 등록 이유

- 고객의 요구에 의해
- 버그가 발견되었을 때
- 사용자가 직접 Issue를 등록할 때

#### 2) Issue를 등록하는 주체

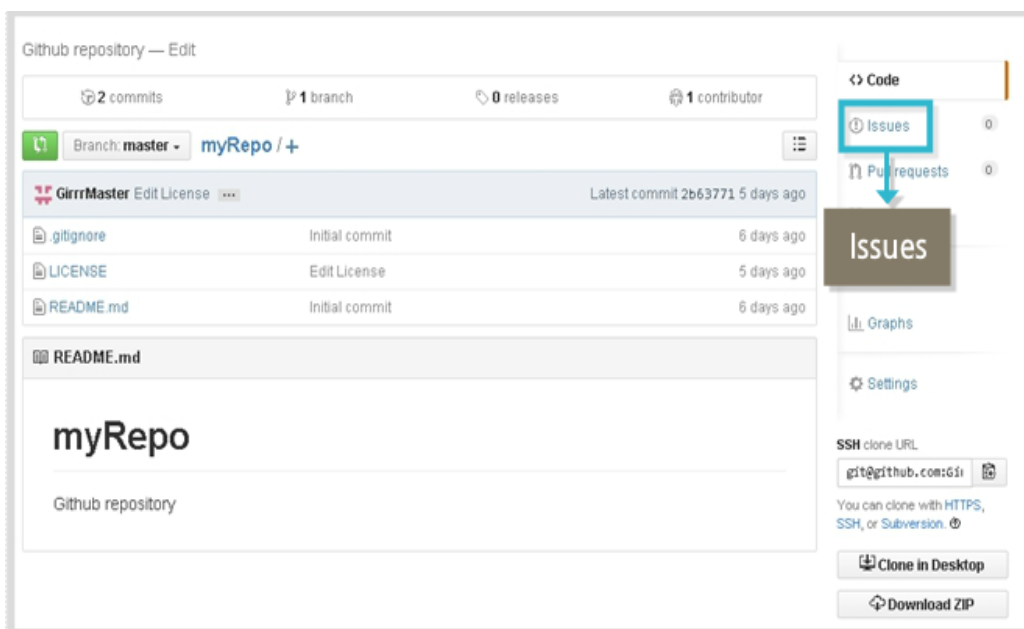
- 기업(고객)
  - 고객과의 접점에 있는 부서에서 고객의 요청을 통해 입력
  - 개발을 총괄하는 프로젝트 리더가 직접 Issue를 등록
- 개발자
  - 개발 중 발생한 문제가 내 개발부분이 아닐 경우 다른 개발자에게 알리기 위해
  - 내가 개발하는 부분에서도 향후에 빼먹지 않고 처리를 위해



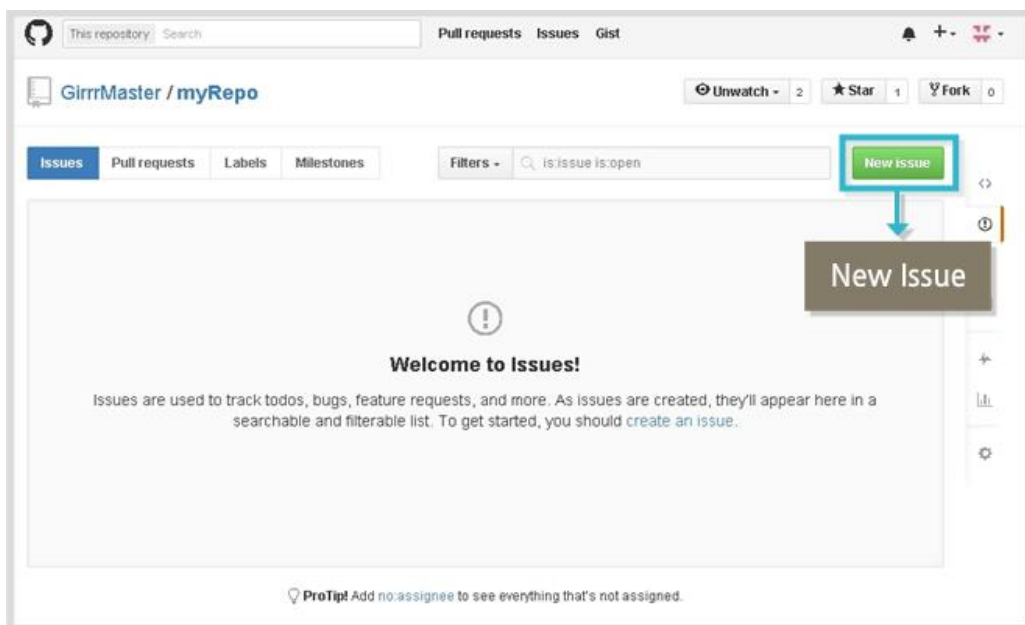
## GitHub의 Issue 관리

### 2. Issue 생성

#### 3) Repository화면 Issue 클릭



#### 4) New Issue 클릭





## GitHub의 Issue 관리

### 2. Issue 생성

#### 5) Issue의 제목, 내용 입력

Issues Pull requests Labels Milestones

첫번째 issue 입니다

Write Preview Styling with Markdown is supported

issue 생성합니다.

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Submit new issue

#### 6) Preview 제공

Issues Pull requests Labels Milestones

첫번째 issue 입니다

Write Preview Styling with Markdown is supported

issue 생성합니다.

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Submit new issue



## GitHub의 Issue 관리

### 2. Issue 생성

#### 7) Submit new issue 클릭

Issues Pull requests Labels Milestones

첫번째 issue 입니다

Write Preview Styling with Markdown is supported

issue 생성합니다.

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Submit new issue

#### 8) Issue 수정

This repository Search Pull requests Issues Gist

GirrrMaster / myRepo

Unwatch 2 Star 1 Fork 0

첫번째 issue 입니다 #1

Open GirrrMaster opened this issue just now · 0 comments

GirrrMaster commented just now Owner

issue 생성합니다.

Write Preview Styling with Markdown is supported

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Close issue Comment

Edit New issue

Labels None yet

Milestone No milestone

Assignee No one—assign yourself

Notifications Unsubscribe

You're receiving notifications because you authored the thread.

1 participant

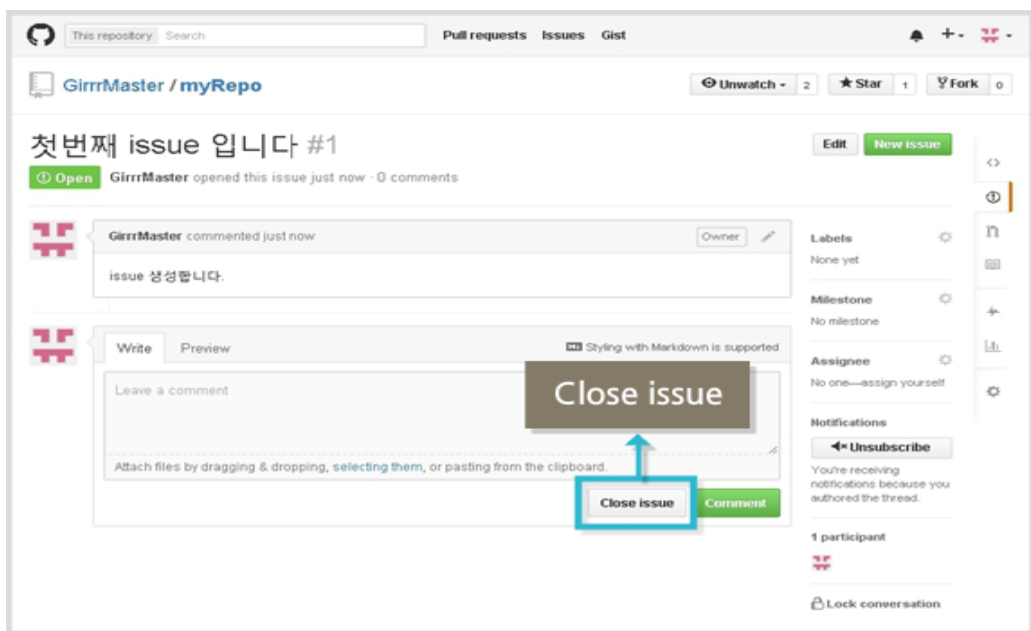
Lock conversation



## GitHub의 Issue 관리

### 2. Issue 생성

#### 9) Repository에 참여자 간 토론



- 다양한 댓글 등을 통해 Issue 해결을 위한 토론
- 새로운 사람을 불러오거나 담당자를 지정하거나 새로운 일정을 만들거나 하는 등의 작업

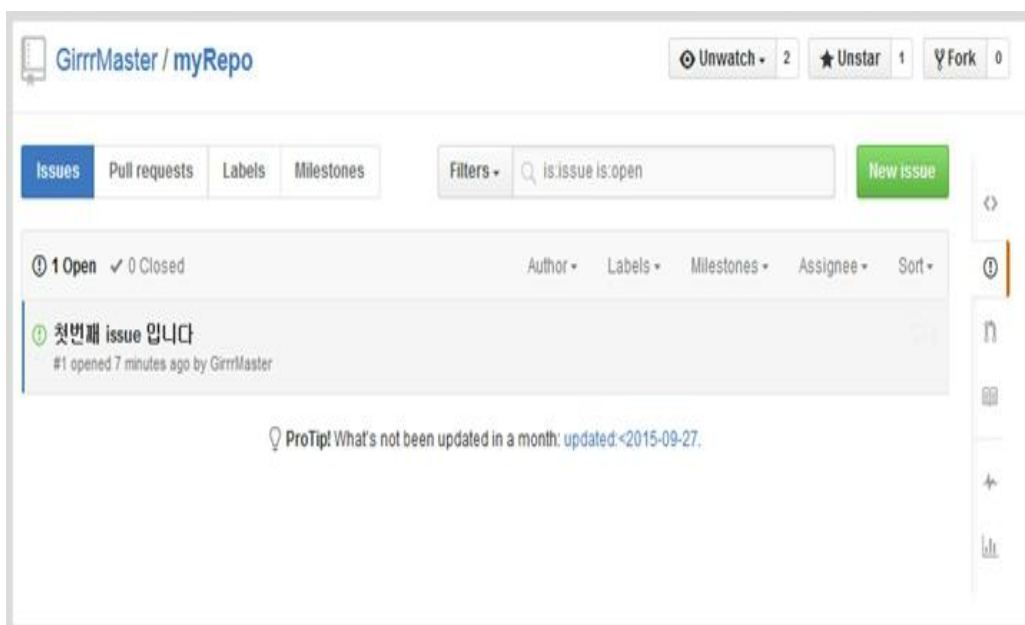




## GitHub의 Issue 관리

### 3. Issue 관리

#### 1) 해당 Issue 클릭



#### 2) Comment 추가

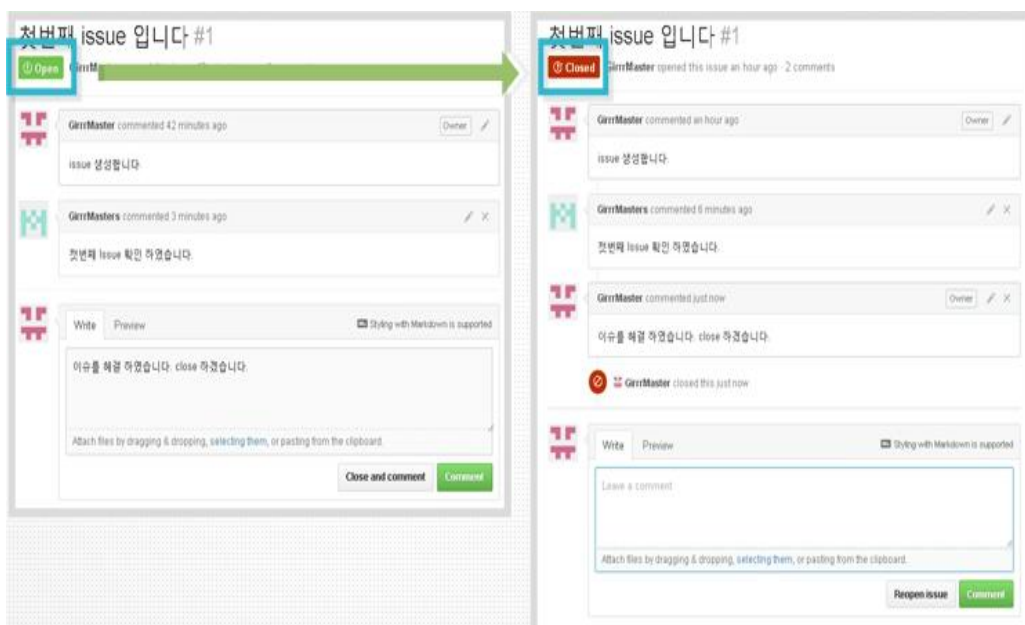




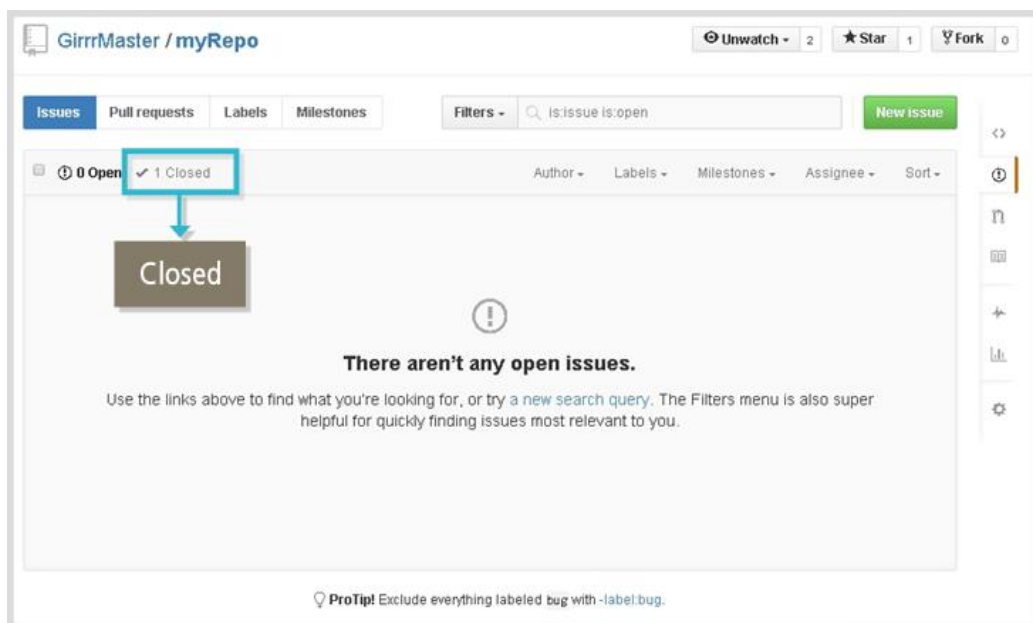
## GitHub의 Issue 관리

### 3. Issue 관리

#### 3) Issue close



#### 4) Issue확인

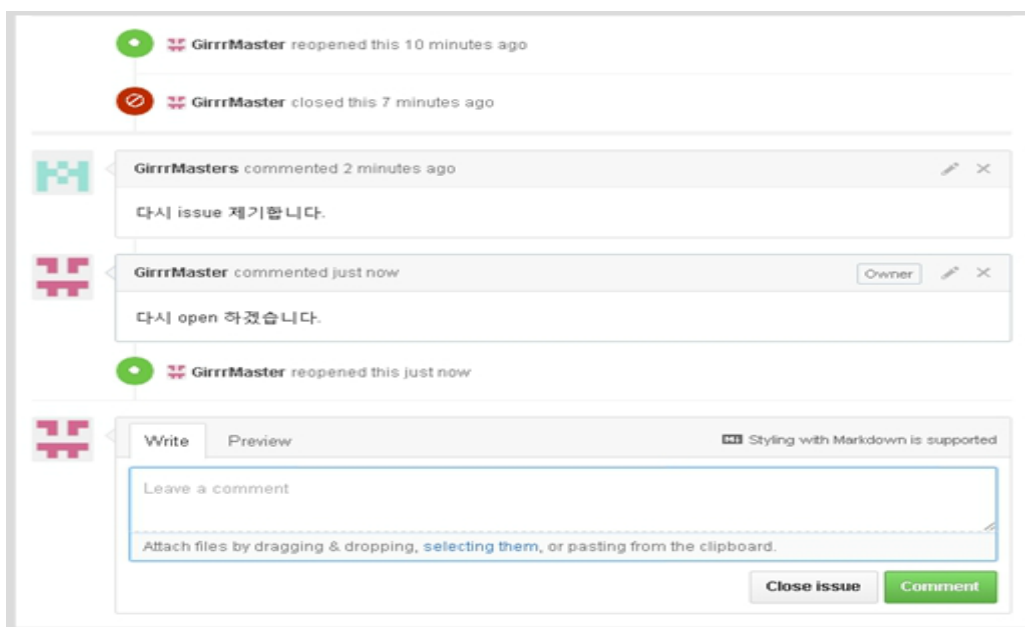




## GitHub의 Issue 관리

### 3. Issue 관리

#### 5) Reopen

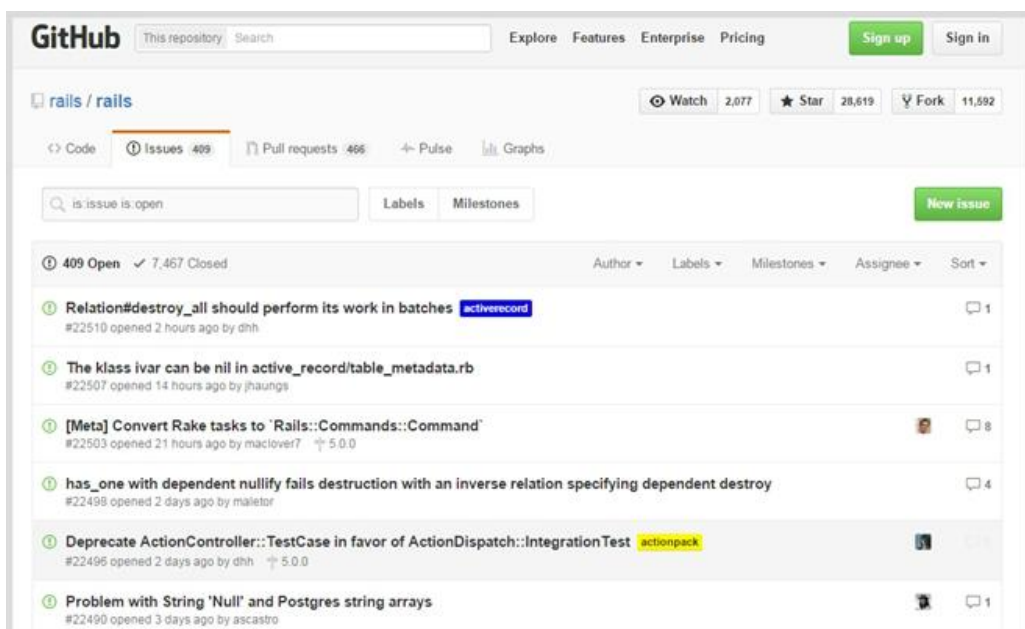




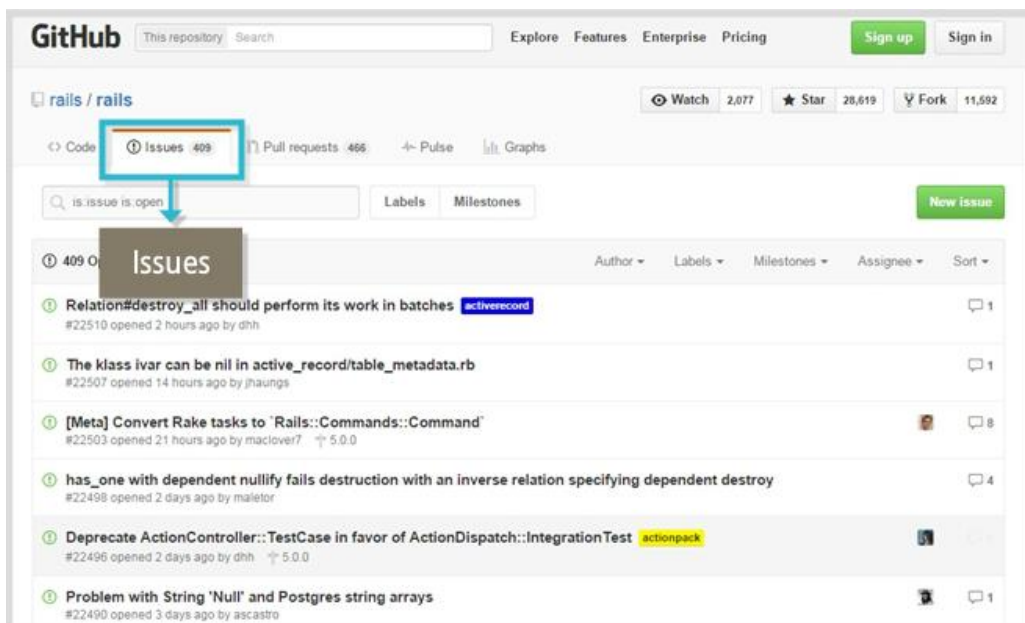
## GitHub의 Issue 관리

### 4. Issue 관리 예제

#### 1) Rails 프로젝트에 접속



#### 2) 상단의 Issue 클릭

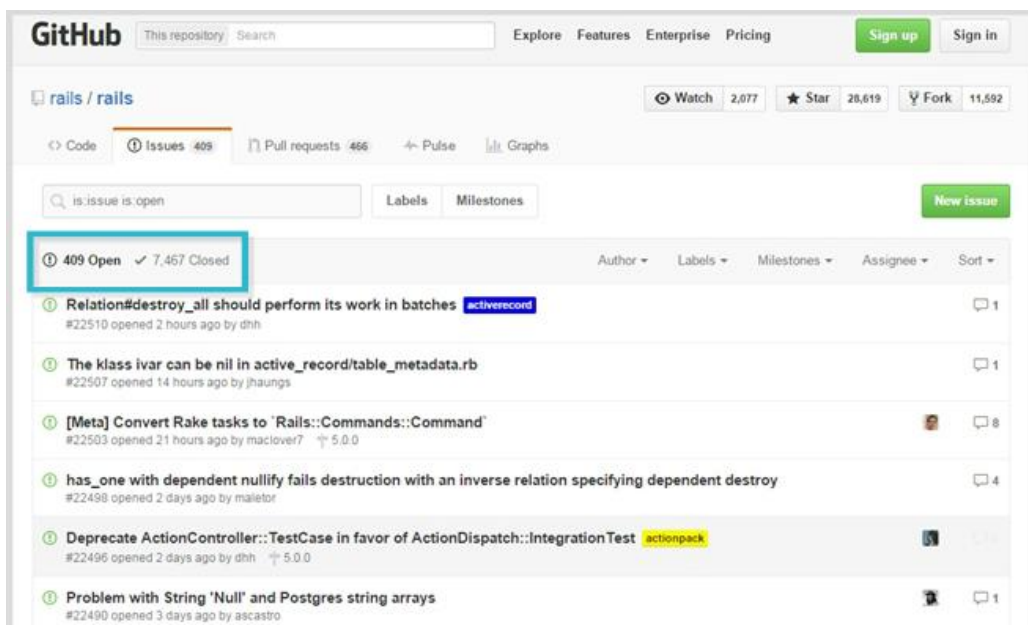




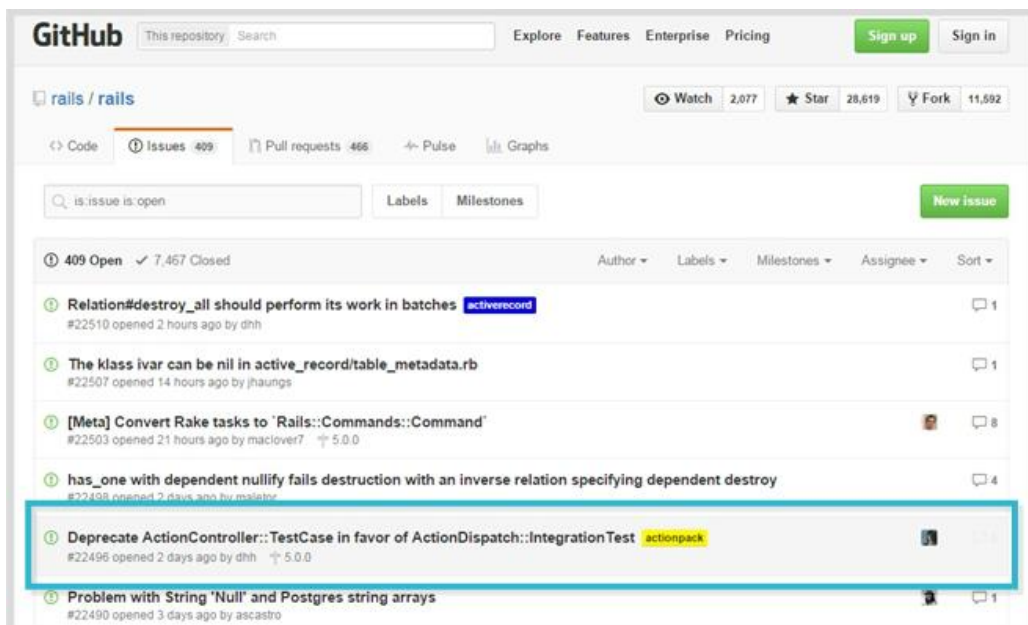
## GitHub의 Issue 관리

### 4. Issue 관리 예제

#### 3) Issue 리스트 확인



#### 4) 확인하고자 하는 Issue 클릭

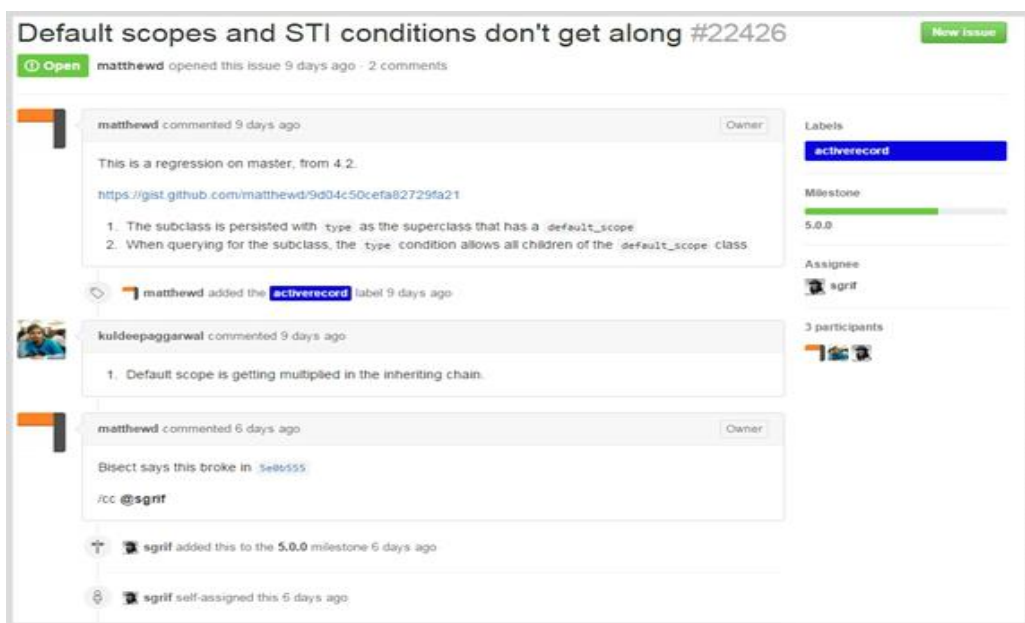




## GitHub의 Issue 관리

### 4. Issue 관리 예제

#### 5) Issue확인



- GitHub의 Issue 관리 시스템을 이용해서 간단하면서도 훌륭하게 프로젝트의 Issue 관리



## ! 핵심정리



### 프로젝트에서의 Issue 관리

#### 1. Issue 관리 방법

- 모든 Issue에 대해 무조건 등록 - 지속적인 관리 추적 가능
- 고객 등의 참여 유도 - 프로젝트 확인 및 수정사항에 대해 즉각적인 피드백 가능
- 개발자가 직접 기간을 산정, PM이 향후 보완을 해 주는 방향으로 정리
- 고객에게도 Issue관리 시스템을 수시로 체크하도록 권장
- 관리자는 Issue를 일일 단위로 관리하여 그 수행 결과를 Issue관리 시스템에 반영

#### 2. Issue 관리 팁

- 좋은 라벨 사용
- 일의 종류 분류
- 난이도 설정
- 우선순위 라벨링

## ! 핵심정리



### GitHub의 Issue 관리

#### 1. Issue 화면 구성

- Issue 시스템 접속 → 새로운 Issue 생성 → Issues 메뉴 추가

#### 2. Issue 생성

- Repository화면 Issue 클릭 → New Issue 클릭 → Issue의 제목, 내용 입력 → Preview 제공 → Submit new issue 클릭 → Issue 수정 → Repository에 참여자 간 토론

#### 3. Issue 관리

- 해당 Issue 클릭 → Comment 추가 → Issue close → Issue 확인 → Reopen

#### 4. Issue 관리 예제

- Rails 프로젝트에 접속 → 상단의 Issue 클릭 → Issue 리스트 확인 → 확인하고자 하는 Issue를 클릭 → Issue확인