

Git & STS project

오류 정리



목차

1. 기본 Git 주의점

- Git 폴더 중첩 구조
- Clone 폴더 경로 확인
(Mac, Window)

2. STS project metadata 오류

- Metadata란?
- STS metadata
- Sts의 git 설정

3. Branch

- Branch 기능
- 사용 의도

01 기본 Git 주의점

- Git 폴더
- Git 폴더 중첩 에러
- Clone 폴더 경로 확인 (Mac, Window)

Git 폴더

Git 폴더 (.git)

- **.git은 Git 저장소의 모든 정보(이력, 설정, 브랜치 등)를 담고 있는 숨겨진 시스템 폴더입니다.**
- **이 폴더가 존재해야 Git 명령어들이 작동하며, Git은 이 폴더가 있는 곳을 기준으로 "이 프로젝트는 Git으로 관리 중이다"라고 판단합니다.**
- **.git 폴더를 삭제하면 Git 저장소가 해제됨 (이력도 사라짐)**
- **Git으로 관리하고 싶지 않은 폴더에는 .git이 없어야 함**
- **다른 .git 폴더 안에 또 .git 폴더가 들어가면 충돌이나 오류 발생 가능**

Git 폴더

Create a new repository

Name

gitFolder

Description

Local path

D:\

Choose...

☐ Initialize this repository with a README

Git ignore

None

License

None

The repository will be created at

D:\gitFolder

Create repository

Cancel

내 PC > 로컬 디스크 (D:) > gitFolder >				
📁 📄 🗑️ ⬆️⬆️ 정렬 ▾ ≡ 보기 ▾ ⋮				
이름	수정한 날짜	유형	크기	
📁 .git	2025-07-08 오후 4:39	파일 폴더		
⚙️ .gitattributes	2025-07-08 오후 4:39	Git Attributes 원...	1KB	

Git 폴더 중첩 에러

Git 폴더 (.git)

- | 항목 | 설명 |
|-----|--------------------------------|
| 원인 | .git 폴더가 중첩되거나 복사됨 |
| 증상 | Git이 저장소를 혼동하거나 무시함 |
| 해결 | 하위폴더의 .git 삭제하거나 서브모듈로 처리 |
| 방지법 | git clone이나 복사 시 .git 포함 여부 주의 |

```
/gitFolder/  
├── .git/      ← 바깥 Git 저장소  
└── gitFolder2/  
    ├── .git/  ← 또 다른 Git 저장소 (중첩됨)
```

Git 폴더 중첩 에러

Create a new repository

Name

gitFolder2

Description

Local path

D:\gitFolder

Choose...

☐ Initialize this repository with a README

Git ignore

None

License

None

The repository will be created at

D:\gitFolder\gitFolder2.

Create repository

Cancel

내 PC > 로컬 디스크 (D:) > gitFolder >

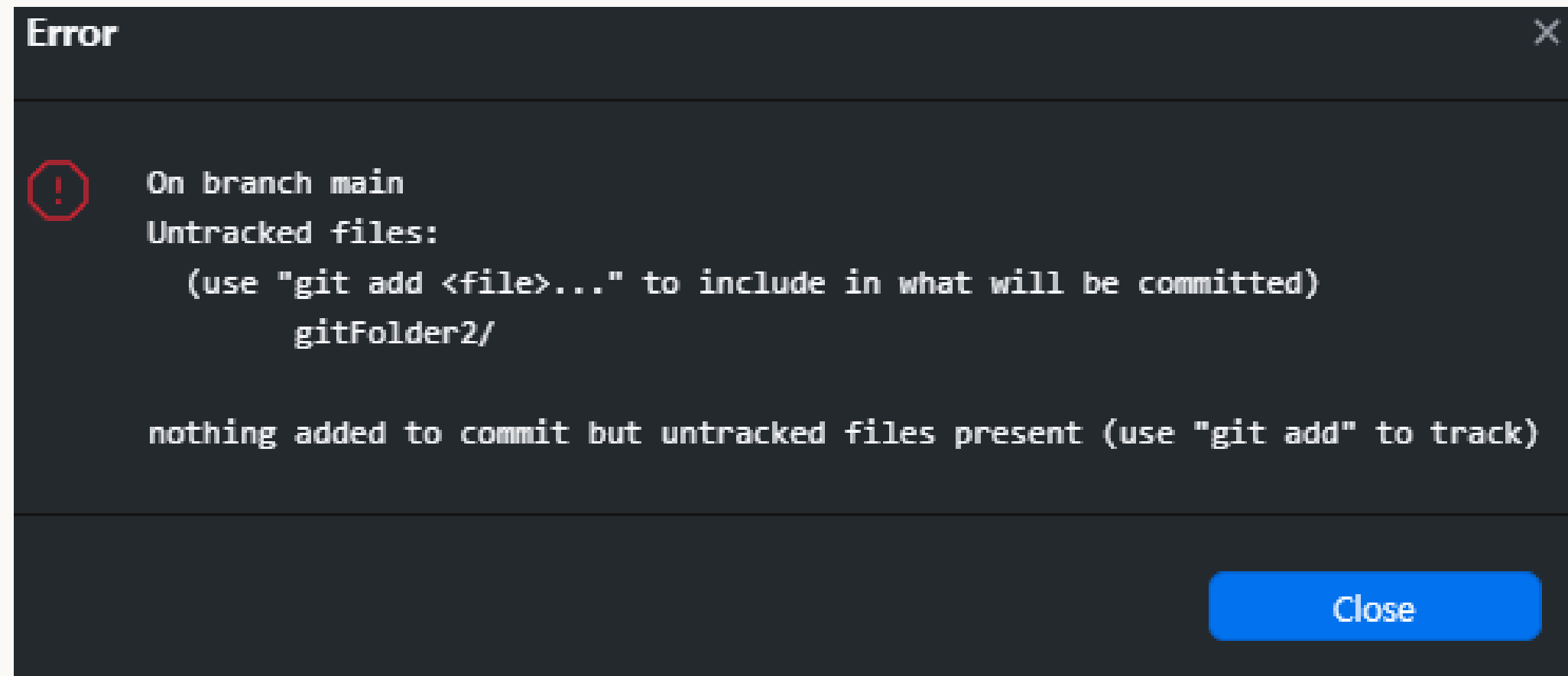
이름	수정한 날짜	유형	크기
.git	2025-07-08 오후 4:39	파일 폴더	
gitFolder2	2025-07-08 오후 4:42	파일 폴더	
.gitattributes	2025-07-08 오후 4:39	Git Attributes 원...	1KB

기본 Git으로 관리하는 폴더 안에 또 Git을 추가하면 충돌이 일어납니다.

내 PC > 로컬 디스크 (D:) > gitFolder > gitFolder2 >

이름	수정한 날짜	유형	크기
.git	2025-07-08 오후 4:42	파일 폴더	
.gitattributes	2025-07-08 오후 4:42	Git Attributes 원...	1KB

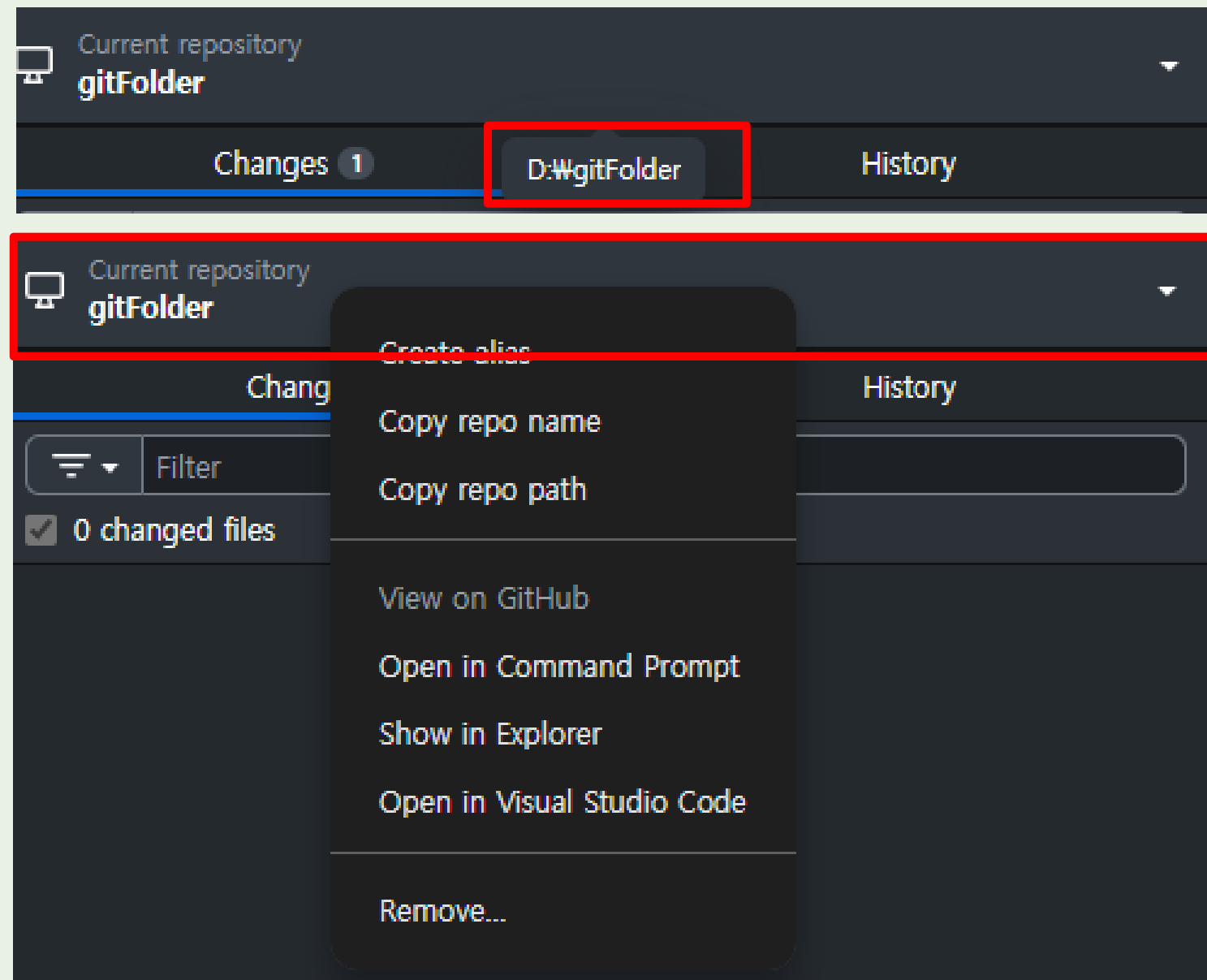
Git 폴더 중첩 에러



상위 폴더 (gitFolder)에서 커밋을 진행 시 하위 폴더(gitFolder2)을
추적(찾기)할 수 없어 나는 에러
(이유는 .git 폴더가 중첩 되어서)

Clone 폴더의 경로 확인 방법

Github desktop 확인

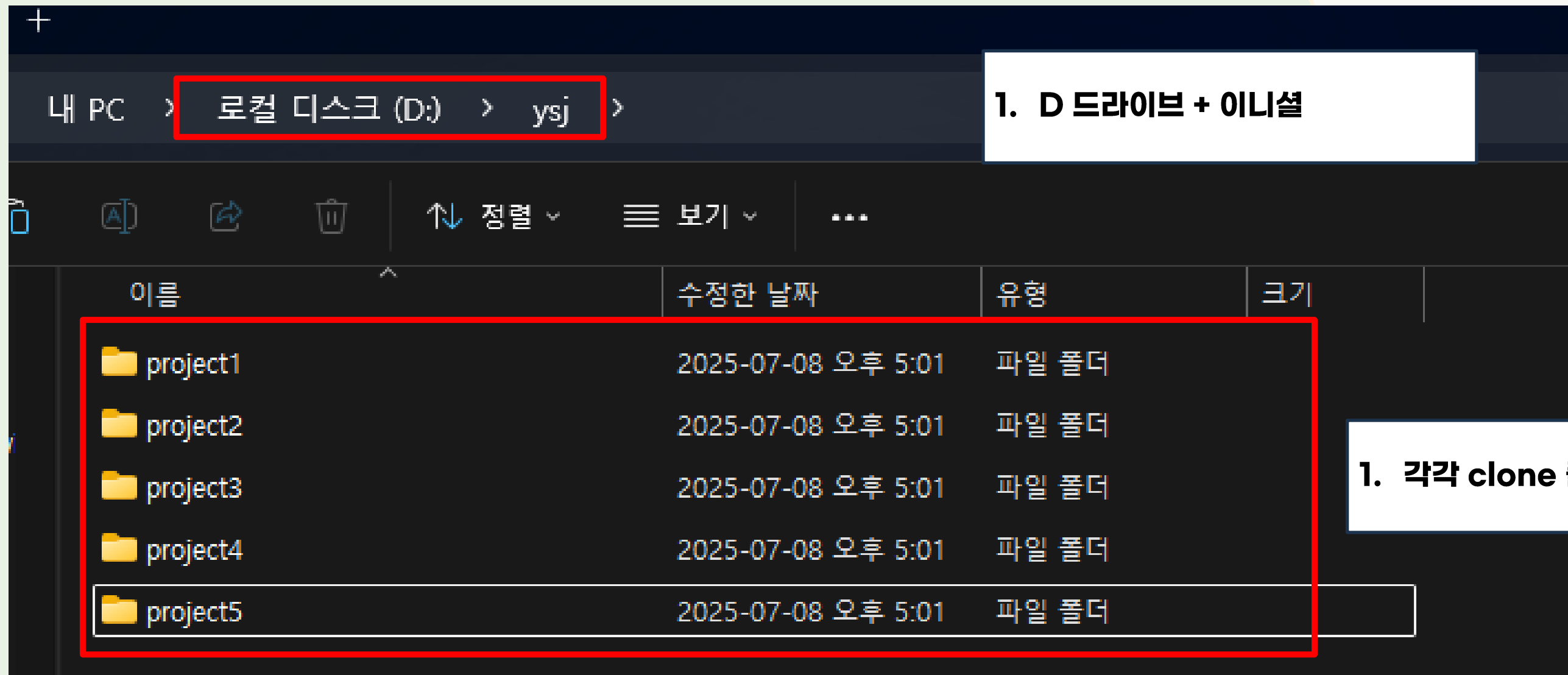


1. Current repository 마우스 올리기 (클릭 x) 오른쪽 마우스 클릭
2. Window : Show in Explorer
Mac : Open in Finder
3. 자신이 clone 혹은 git 연동 폴더의 경로는 기억하는 습관 기르기

Window 추천 Clone 경로

1. D 드라이브
2. D 드라이브에 만 생성
3. 혹은 D 드라이브에 자기 이니셜 폴더 생성 후 그
폴더에서 만 관리
(중첩 구조 X 따로따로 생성 관리)

Window 추천 Clone 경로



1. D 드라이브 + 이니셜

1. 각각 clone 폴더 입니다.

Mac 추천 Clone 경로

```
/Users/ysj/  
├─ Desktop/      ← 바탕화면  
├─ Documents/    ← 문서  
├─ Downloads/    ← 다운로드  
├─ Music/        ← 음악  
├─ Pictures/     ← 사진  
├─ Movies/       ← 동영상  
├─ Applications/ ← 사용자가 설치한 앱  
├─ Library/      ← 사용자 전용 설정 파일 (숨겨짐)  
└─ Projects/     ← (직접 만든 폴더, 예: 개발 폴더)
```

```
/Users/ysj/  
├─ Projects/  
│   └─ my-clone-repo1/  
│       └─ my-clone-repo2/
```

1. 클론 프로젝트 모음 폴더
(직접 생성 후 clone 경로로 활용)

02 STS project metadata 오류

- Metadata란?
- STS metadata
- Sts의 git 설정

Metadata 란?

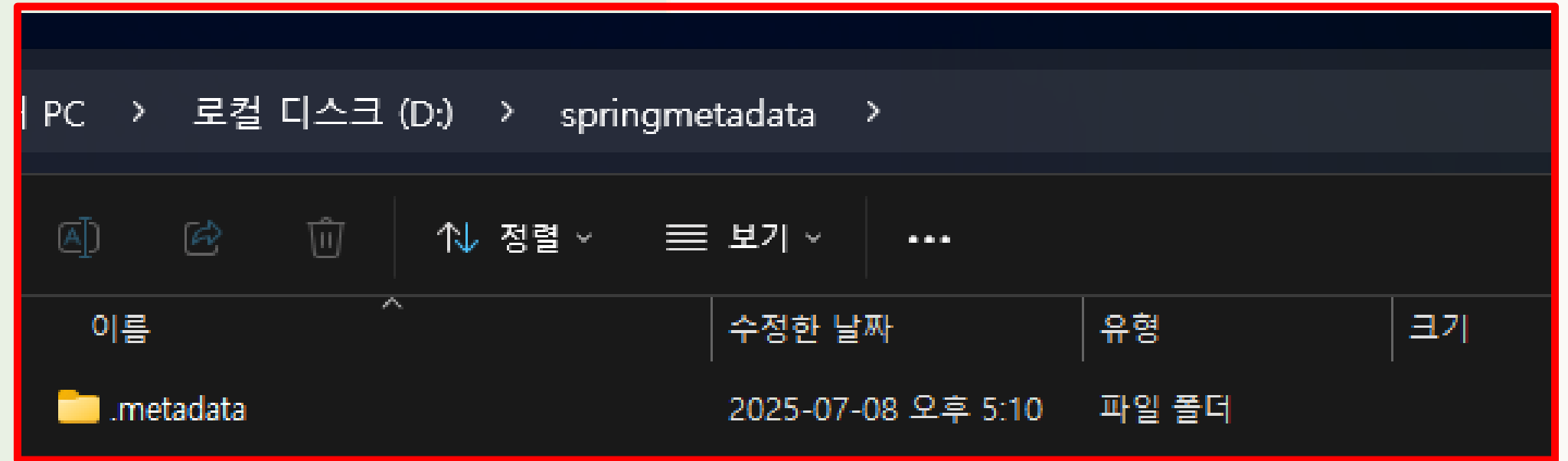
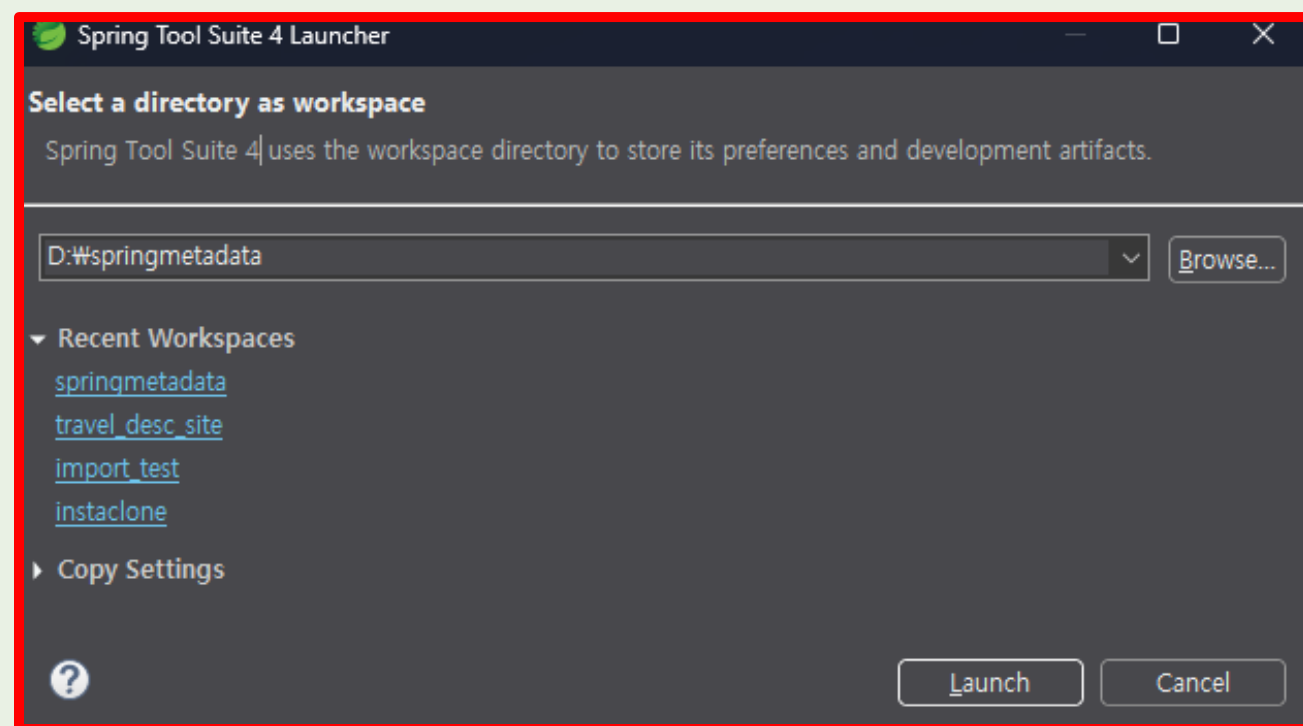
- "데이터에 대한 데이터", 즉 어떤 데이터의 성질, 구조, 설명 등을 담고 있는 정보입니다.
- 데이터를 설명하는 정보

실제 데이터	메타데이터 예시
사진 파일 (photo.jpg)	파일명, 용량, 해상도, 촬영일, 카메라 정보
엑셀 파일	시트 수, 마지막 수정일, 작성자
노래 파일 (.mp3)	제목, 가수, 앨범, 재생 시간
데이터베이스 테이블	컬럼명, 자료형, 기본키 여부, 제약조건

STS metadata

STS 에서 metadata 생성 시점

- Workspace(워크 스페이스) 지정 해당 경로 폴더에 metadata가 생성



STS에서의 metadata가 담고 있는 정보는?

```
.metadata/  
├─ .log           ← 에러 로그 파일  
├─ .plugins/      ← 플러그인 설정 정보  
│   └─ org.eclipse.m2e/      ← Maven 설정  
│   └─ org.eclipse.core.resources/ ← 워크스페이스 파일 구조  
│   └─ org.eclipse.jdt.core/  ← Java 설정 관련  
├─ .lock          ← 워크스페이스 동시 실행 방지  
├─ workspace.xml   ← 기본 설정 정보  
└─ version.ini     ← Eclipse 버전 정보
```

- .metadata는 STS(또는 Eclipse)가 처음 워크스페이스를 만들 때 한 번 생성되며, 이후 사용자가 "무언가를 추가하거나 변경"할 때마다 그 내부 정보가 계속 갱신됩니다.

에러 이유

상황 설명

- Main에서 기존 틀을 그대로 Branch로 가져옴 (에러 X)
- STS 에서 의존성(라이브러리)를 추가함 -> 메타데이터 생성
- 현재 STS를 키고 있는 중에 main으로 돌아가면
메타데이터 폴더에 생성된 파일이 삭제를 해야하나 STS에서 쓰고(점유)있는 상태여서
에러 발생
- 메타데이터는 생성되거나 삭제되거나 하는 작업이 많이 생긴다.

에러 이유

상황 설명

Error

!

```
warning: in the working copy of '.metadata/.log', LF will be replaced by CRLF the next time
Git touches it
warning: in the working copy of
'.metadata/.plugins/org.eclipse.jdt.core/savedIndexNames.txt', LF will be replaced by CRLF
the next time Git touches it
warning: in the working copy of 'demo/pom.xml', LF will be replaced by CRLF the next time Git
touches it
Saved working directory and index state On error-test: !!GitHub_Desktop<error-test>
error: unable to unlink old '.metadata/.plugins/org.eclipse.m2e.logback/0.log': Invalid
argument
fatal: Could not reset index file to revision 'HEAD'.
```

Close

해결 방법

.metadata를 .gitignore 추가 혹은 STS 종료

그러나 근본적인 원인을 .gitignore 처리를 해줘야 한다.

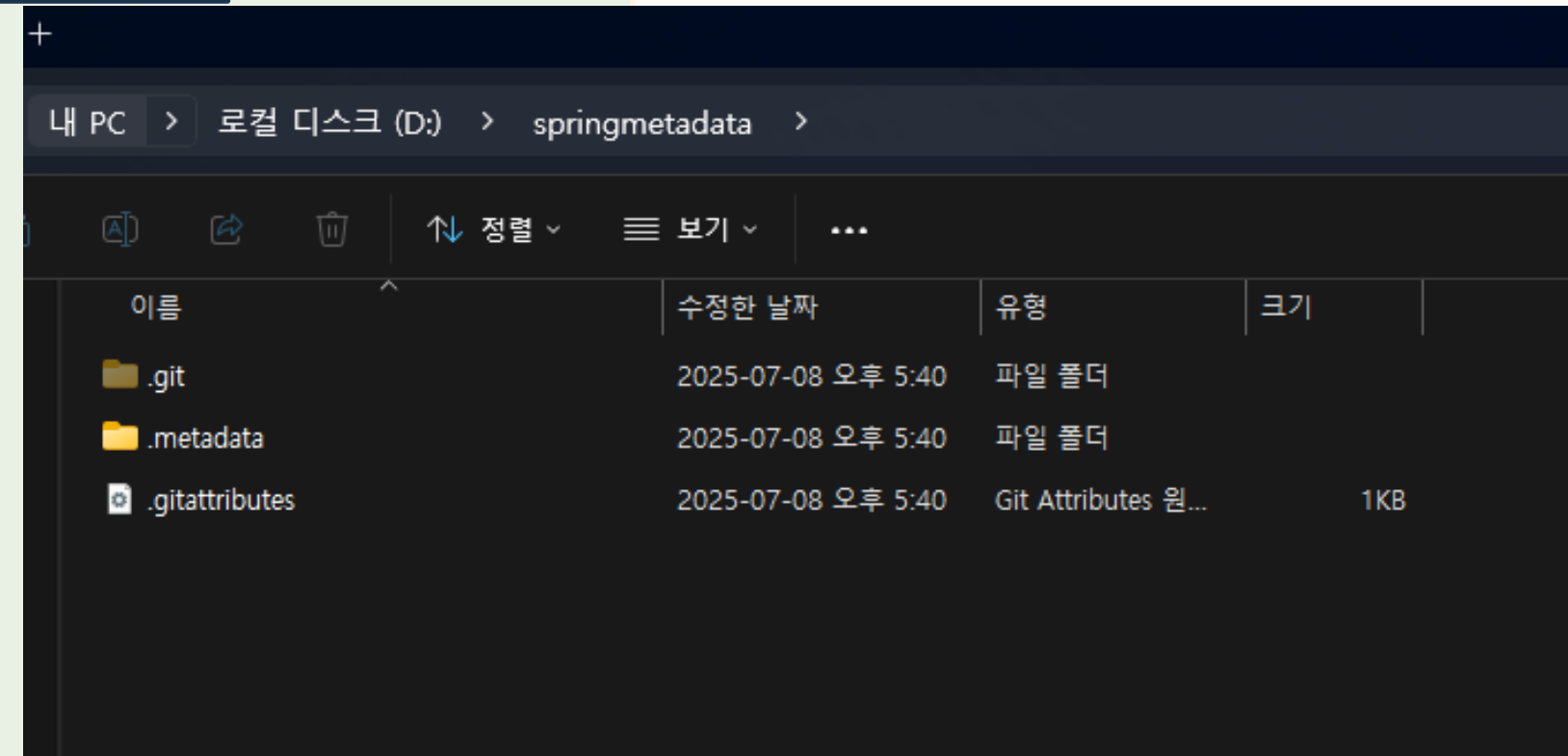
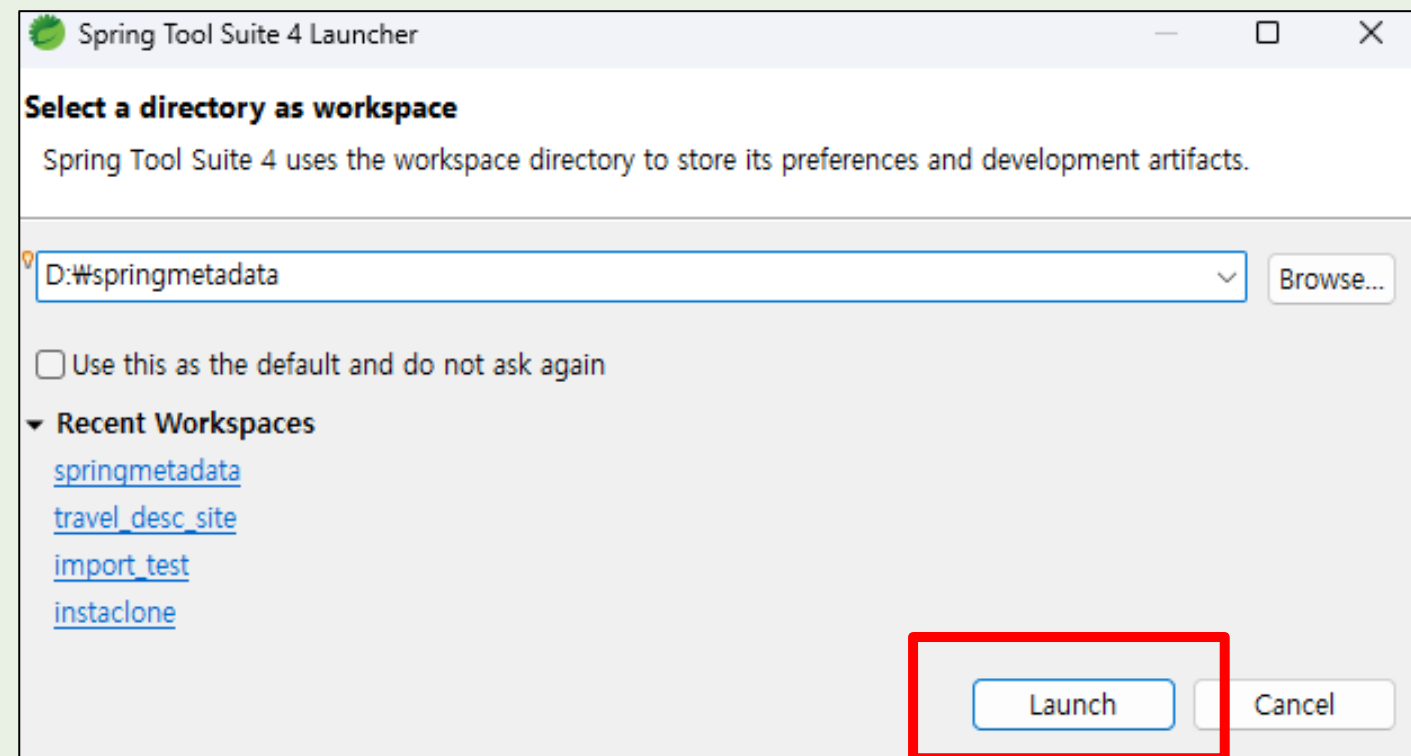
이유 : metadata는 데이터에 대한 정보 ! 즉 현재 STS에서 작업하는 개발 설정들이 들어있으므로 제외 시키는 것이 맞다.

다른 사람이 해당 .metadata를 받아도 제대로 작동하지 않으며, 오히려 충돌이나 오류 유발 가능

.metadata 안의 파일은 자주 바뀌고 Git 입장에서 "불필요한 잡음"이 됨

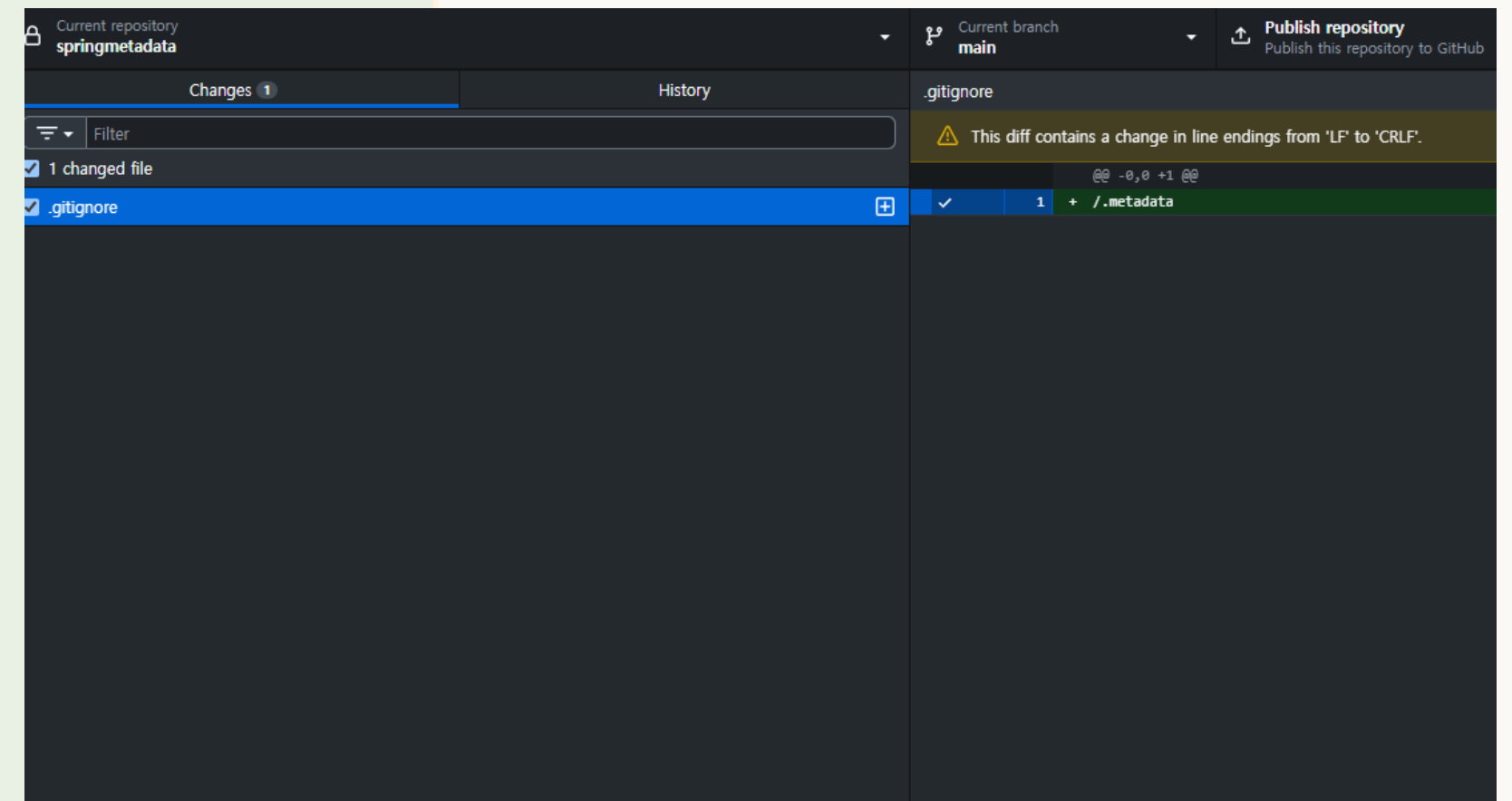
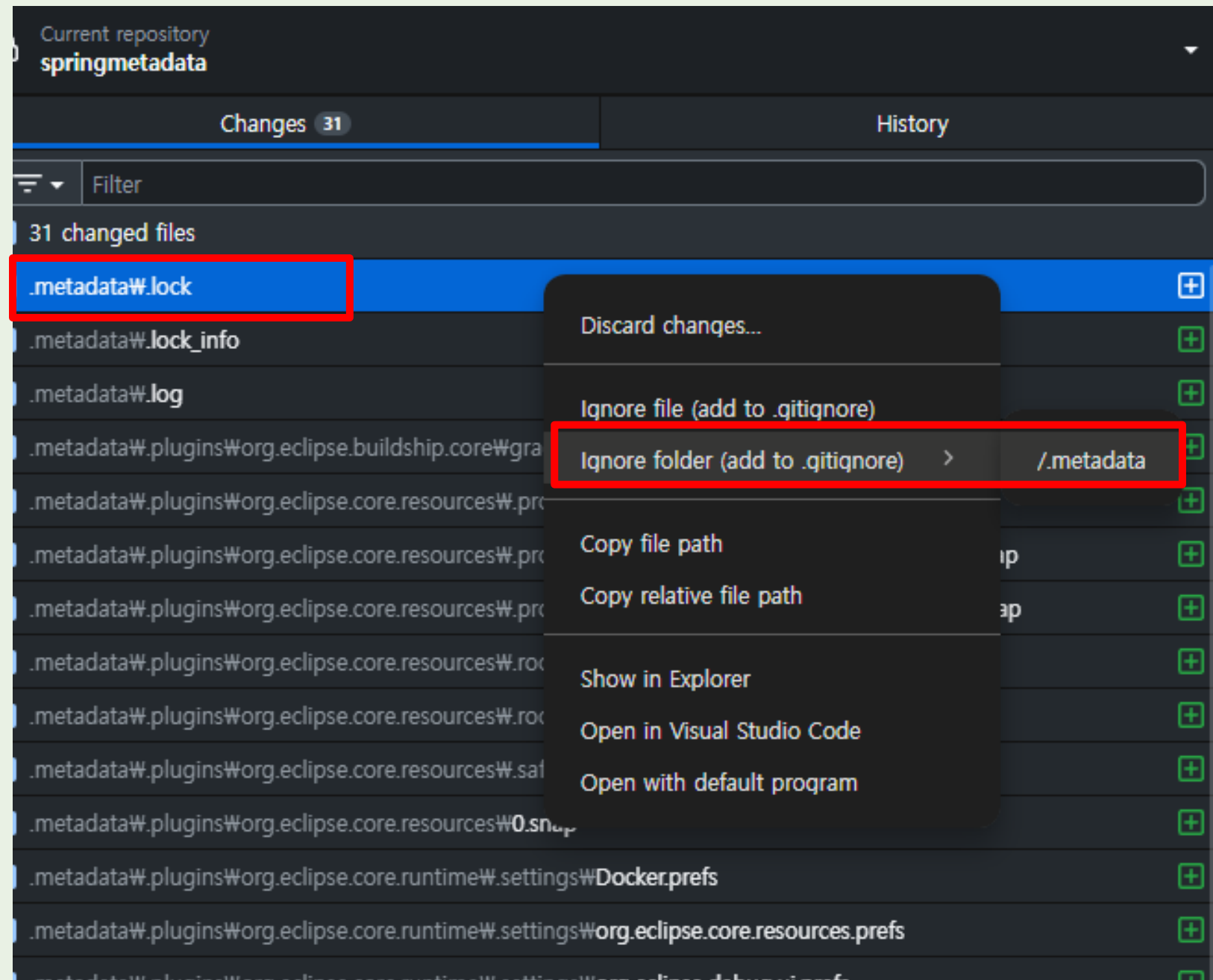
.gitignore 추가

워크스페이스를 지정하고 초기에 .gitignore를 추가하는 걸 추천



.gitignore 추가

1. Metadata 글씨가 써있는 파일 하나 클릭 후 오른쪽 마우스 클릭
2. Ignore folder -> /.metadata



요약 정리

- ⚠ Git 사용 시 주의사항 요약
 1. ✅ Git 폴더는 중첩 금지!
 2. .git 폴더가 또 다른 .git 안에 존재하면 Git이 제대로 작동하지 않거나 충돌 발생 가능
 3. Git 저장소는 하나의 루트 폴더 기준으로만 작동해야 함
- ✅ .metadata는 Git에 포함 ❌
 - .metadata는 STS의 워크스페이스 설정 정보로, 코드와 직접적인 관련이 없는 개인 개발 환경 데이터입니다.
 - Git에 포함되면 다른 사람의 환경에서 오류, 충돌, 실행 불가 문제가 발생할 수 있음
 - 내부 파일은 자주 변경되며, Git 입장에서 쓸모없는 노이즈가 됩니다.

03 Branch

- Branch 기능
- 사용 의도

Branch 기능

목적	설명
독립적인 작업 공간 생성	현재(main) 코드를 유지한 채, 새로운 기능 개발이나 수정 작업을 분리해서 수행
안정성 확보	실험적인 변경 사항이 기존 코드에 영향을 주지 않도록 격리
여러 작업을 병렬로 수행	팀원들이 각자 브랜치에서 기능을 나눠 개발 → 병합(Merge)으로 통합
기능 단위 관리	기능(feature), 버그 수정(bugfix), 배포(release) 등을 명확하게 구분 가능
롤백 및 추적 용이	브랜치를 사용하면 변경 이력을 독립적으로 추적할 수 있어, 되돌리기도 쉬움

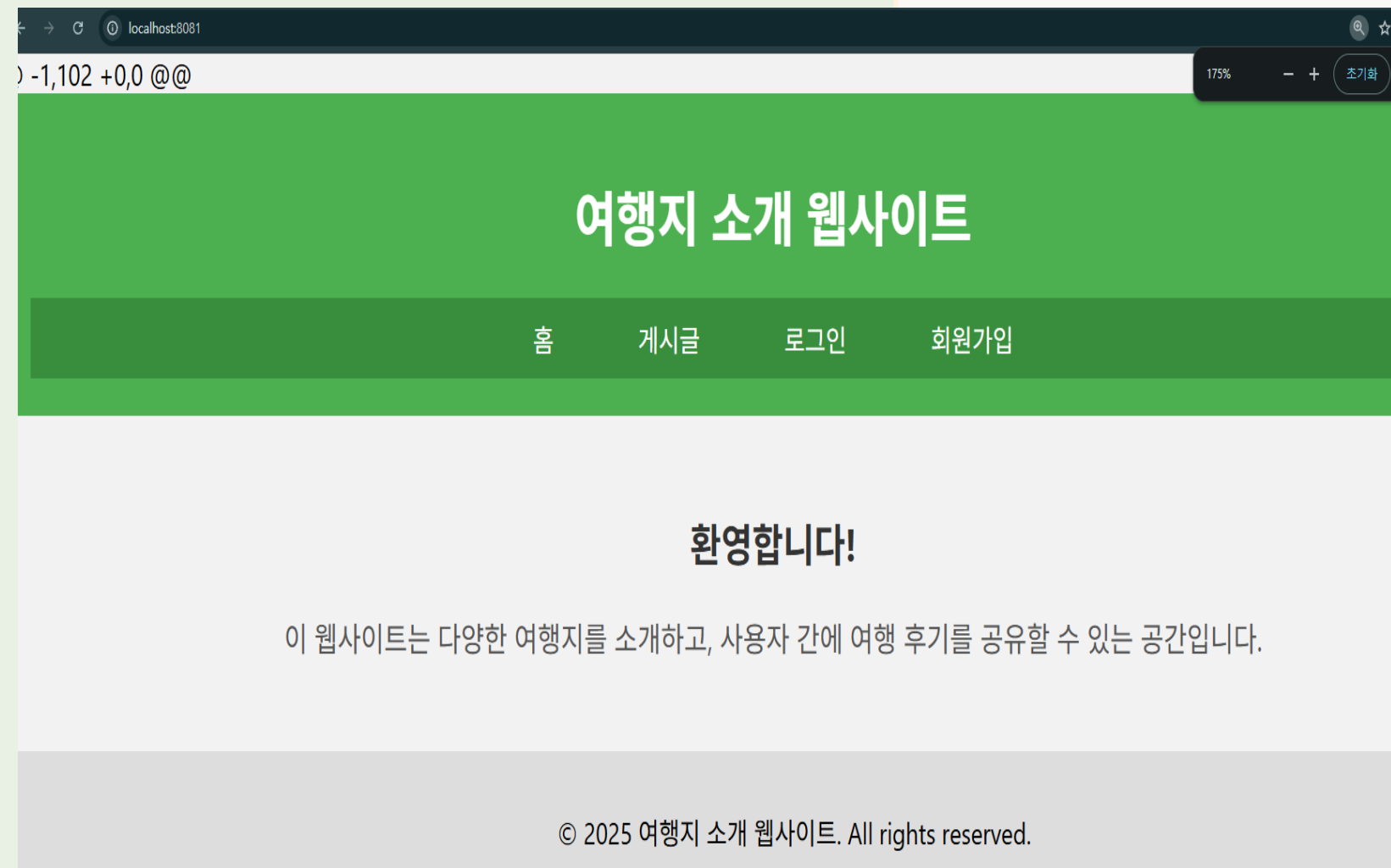
수업 사용 의도

Main branch 코드

```
/* 푸터 스타일 */
footer {
  background-color: #ddd;
  text-align: center;
  padding: 15px;
  position: relative;
  bottom: 0;
  width: 100%;
}
</style>
</head>
<body>
<header>
<h1>여행지 소개 웹사이트</h1>
<nav>
<ul>
<li><a href="#">홈</a></li>
<li><a href="#">게시글</a></li>
<li><a href="#">로그인</a></li>
<li><a href="#">회원가입</a></li>
</ul>
</nav>
</header>

<main>
<h2>환영합니다!</h2>
<p>
  이 웹사이트는 다양한 여행지를 소개하고, 사용자 간에 여행 후기를 공유할
  수 있는 공간입니다.
</p>
</main>

<footer>
<p>&copy; 2025 여행지 소개 웹사이트. All rights reserved.</p>
</footer>
```



수업 사용 의도

jstlpractice branch 코드

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="UTF-8" />
<title>홈페이지 메인</title>
</head>
<body>
<!-- set -->
<c:set var="greeting" value="유승진입니다." />
<p>${greeting}</p>

<c:set var="age" value="19" />

<!-- if문 jstl -->
<c:if test="${age} >= 20">
<p>난 20대 이상이다</p>
</c:if>

<!-- if-else choose문 jstl -->
<c:set var="grade" value="75" />

<c:choose>
<c:when test="${grade} >= 90">
<p>난 A다</p>
</c:when>
<c:when test="${grade} >= 80">
<p>난 B다</p>
</c:when>
<c:otherwise>
<p>난 C다</p>
</c:otherwise>
</c:choose>

<c:set var="subjects" value="${['국어', '수학', '과학']}" />

<c:forEach var="subject" items="${subjects}">
<p>${subject}</p>
</c:forEach>

</body>
</html>
```

localhost:8081

유승진입니다.

난 C다

국어

수학

과학

수업 사용 의도

Main 기준으로 branch 를 다시 하나만들어서 jstlpractice에서 했던 jstl 문법 연습을 다시 하려고 했었습니다.

**Git branch 기능을 사용하지 않고 jsp파일을 새로 하나 만들어서 실습 하면 되나
협업 + 새로운 기능을 테스트할 때 + git 사용 경험 능력을 키울려고 했었습니다.**