

Netty-Starter-With-Spring-Boot (Day1)

프로젝트 개요



개발이란 단어를 들었을 때 보통 많이 생각하시는 게, 보통 웹 개발, 모바일 개발이라고 생각합니다.

하지만 저는 그런 분야 뿐 아닌 제가 담당하는 서버 개발에 대해서도 너무 재밌고 많은 사람들이 관심을 갖고 있지만

친숙하지 않거나, 러닝커브가 높다고 생각하시는 분들을 위해, 현재 많이 사용하는 Java 기반의 비동기 네트워크 프레임워크인

Netty를 통해서 **Netty**의 기본 개념과 사용법에 대해 설명 및 실습을 통해 서버 개발에 한 걸음 더 친숙해질 수 있도록 도와드리겠습니다.


오늘의 목표

1. Spring Boot 프로젝트를 생성할 수 있다.
2. GitHub Repository를 생성할 수 있다.
3. Spring Boot 프로젝트를 원격 저장소에 연결할 수 있다.
4. 원격 저장소에 파일을 업로드 할 수 있다.

1. Spring Boot Project Created

1.1 start.spring.io 링크 접속

- IntelliJ Ultimate가 아닌 경우 해당 사이트에서 만들어야 합니다.
- Ultimate인 경우 프로젝트 생성에서 만들 수 있습니다.
- 접속 후에는 아래와 같은 창이 나옵니다.



Project
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Maven**

Language
☒ **Java** ☐ Kotlin ☐ Groovy

Spring Boot
☐ 3.3.2 (SNAPSHOT) ☒ **3.3.1** ☐ 3.2.8 (SNAPSHOT) ☐ 3.2.7

Project Metadata
 Group
 Artifact
 Name
 Description
 Package name
 Packaging ☒ **Jar** ☐ War
 Java ☐ 22 ☐ 21 ☒ **17**


Dependencies ADD DEPENDENCIES... CTRL + B
 No dependency selected


GENERATE CTRL + G EXPLORE CTRL + SPACE SHARE...

1.2 Maven Or Gradle

- Maven, Gradle 둘 중에 하나를 선택하시면 됩니다.
- 저는 해당 프로젝트에서는 Maven으로 진행 하겠습니다.

1.3 Spring Boot Version & JDK Version



[Why Spring](#) [Learn](#) [Projects](#) [Academy](#) [Solutions](#) [Community](#) 

Spring blog [All Posts](#) [Engineering](#) [Releases](#) [News and Events](#) RSS feeds

Preparing for Spring Boot 3.0

ENGINEERING | PHIL WEBB | MAY 24, 2022 | 69 COMMENTS

Spring Boot 2.0 was the first release in the 2.x line and was published on February 28th 2018. We've just released Spring Boot 2.7 which means that, so far, we've been maintaining the 2.x line for just over 4 years. In total we've published 95 distinct releases over that timeframe!

The entire Spring team, and many in our community of contributors, are now preparing for the next generation of Spring. We are planning to release Spring Boot 3.0 in November 2022. This next major revision will be based on Spring Framework 6.0 and will require Java 17 or above. It will also be the first version of Spring Boot that makes use of Jakarta EE 9 APIs ([Jakarta.*](#)) instead of EE 8 ([javax.*](#)).

The next six months offer an ideal opportunity to prepare your own projects for this major release. In this blog post we'll cover some of the things that you can do today to make any future migration as painless as possible.

Get the Spring newsletter

Stay connected with the Spring newsletter

SUBSCRIBE

- 기본적으로 Spring Boot 버전이 올라가면서 요구하는 JDK 버전이 올라갔습니다.
- 3.2.7을 선택, JDK 버전은 기본적으로 17을 사용하게 되었습니다.
- 아래는 Spring Boot 공식 사이트에 있는 내용입니다.

- Spring Boot 3.0부터는 JDK 17을 필요로 한다고 적혀있습니다.
 - JDK 11을 사용하고 싶다면 Spring Boot 최신 버전인 2.7.x를 사용하기를 추천합니다.
-

1.4 Group 설정 Ex → com.dev.vihaan

- 보통 기업의 도메인명을 기입하여 사용합니다.
- 개인 프로젝트일 경우 자유롭게 만들면 됩니다.

1.5 Artifact 설정 Ex → netty-starter-with-spring-boot

- 빌드된 결과물 이름을 의미합니다 (.jar)

1.6 Description

- Ex → Netty Practice Project with Spring Boot
- 프로젝트에 대한 설명을 적을 수 있습니다.

1.7 package

- java의 패키지 name을 의미합니다.
-

1.8 Dependency 선택

- 위와 같이 작성을 끝냈다면 오른쪽에서 Dependencies를 클릭하여 Lombok을 선택해 줄 수 있습니다. 단축키 (CTRL + B)
- 설정이 끝났을 때는 아래와 같이 나옵니다. (package 명만 살짝 수정하였습니다.)

The screenshot shows the Spring Initializr web application. The top left has a menu icon and the Spring Initializr logo. Below the logo, there are three tabs: Project, Language, and Dependencies. The Project tab is active, showing options for Maven, Spring Boot 3.2.7, and Java. The Language tab shows options for Java, Kotlin, and Groovy. The Dependencies tab shows a list of dependencies, including Lombok. The Project Metadata section contains fields for Group, Artifact, Name, Description, and Package name. The bottom of the page has three buttons: GENERATE, EXPLORE, and SHARE.

1.9 IntelliJ에서 프로젝트 열기

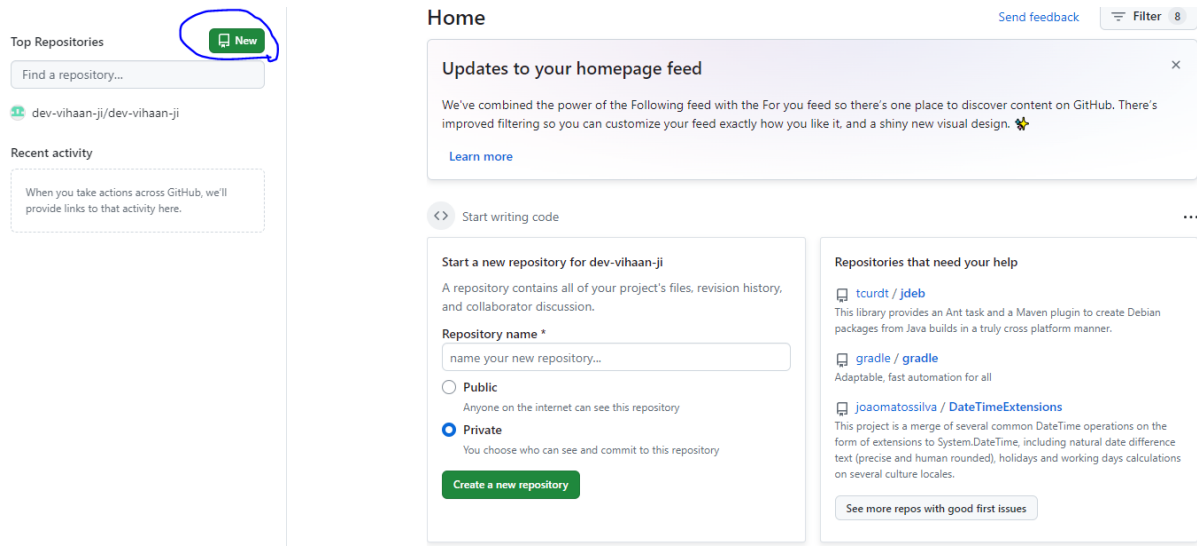
- Generate를 누르면 압축 파일이 나오고 압축을 풀면됩니다.
- 압축을 푼 파일을 IntelliJ를 통해 열 수 있습니다.

2. GitHub Repository Create!

2.1 GitHub login

- GitHub 계정을 통해 로그인을 진행합니다.

2.2 로그인을 한 후 왼쪽 상단에 New 버튼을 클릭



2.3 GitHub Create repository

- 아래 요소들에 대해서는 간략하게 설명만 하겠습니다. 자세한 내용은 찾아서 공부하시면 됩니다.
- **Repository name** : 원격 저장소 이름 (프로젝트 이름)
- **Description (optional)** : 원격 저장소 설명
- **public / private** :
 - **Public**: 누구나 저장소를 볼 수 있고, 복사할 수 있는 공개 저장소입니다.
 - **Private**: 저장소 소유자만 접근할 수 있는 비공개 저장소입니다.
- **README file**
 - 프로젝트에 대한 설명을 담는 파일입니다.
 - 프로젝트의 목적, 기능, 사용 방법, 설치 방법 등을 작성할 수 있습니다.
- **.gitignore**
 - Git이 특정 파일이나 디렉토리를 무시하도록 지정하는 파일입니다.
 - 빌드 결과물, 로그 파일, IDE 설정 파일, 중요한 보안 파일 등 프로젝트에 포함되지 않아야 할 파일들을 지정할 수 있습니다.
 - 프로젝트 시작 시부터 .gitignore 파일을 활용하면 효율적으로 프로젝트를 관리할 수 있습니다.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 dev-vihaan-ji ▾

Repository name *

netty-starter-with-spring-

⚠ The repository netty-starter-with-spring-boot already exists on this account.

Great repository names are short and memorable. Need inspiration? How about [cautious-journey](#) ?

Description (optional)

Project for Netty beginners

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository

3. Spring Boot Project Connect GitHub Repository

3.1 **GitBash** 를 통해 내가 올릴 프로젝트 경로로 접속

- 접속 후 **git init** 명령어를 사용합니다.
- **git init** 이란:
 - **Git**을 로컬 저장소에 생성하기 위해 **Git Bash** 프롬프트를 통해 입력을 하면 **.git** 이라는 파일이 생성됩니다.

- 이 파일은 버전에 대한 정보를 담고 있는 파일이며, 평소에는 숨김 파일로 설정되어 눈에 보이지 않습니다.
- `.git` 파일을 보기 위해서는 `ls -al` 옵션을 주게되면 `.git` 볼 수 있습니다.
- 또한 해당 파일을 삭제하면 이전 버전에 대한 정보가 날아갈 수 있으므로 주의해야 합니다.

```
jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot
$ ls -al
total 33
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:35 ./
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:34 ../
-rw-r--r-- 1 jimin 197121   395 Jun 27 22:34 .gitignore
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:35 .idea/
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:34 .mvn/
-rw-r--r-- 1 jimin 197121  1112 Jun 27 22:34 HELP.md
-rwxr-xr-x 1 jimin 197121 10666 Jun 27 22:34 mvnw*
-rw-r--r-- 1 jimin 197121  6913 Jun 27 22:34 mvnw.cmd
-rw-r--r-- 1 jimin 197121  1766 Jun 27 22:34 pom.xml
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:34 src/

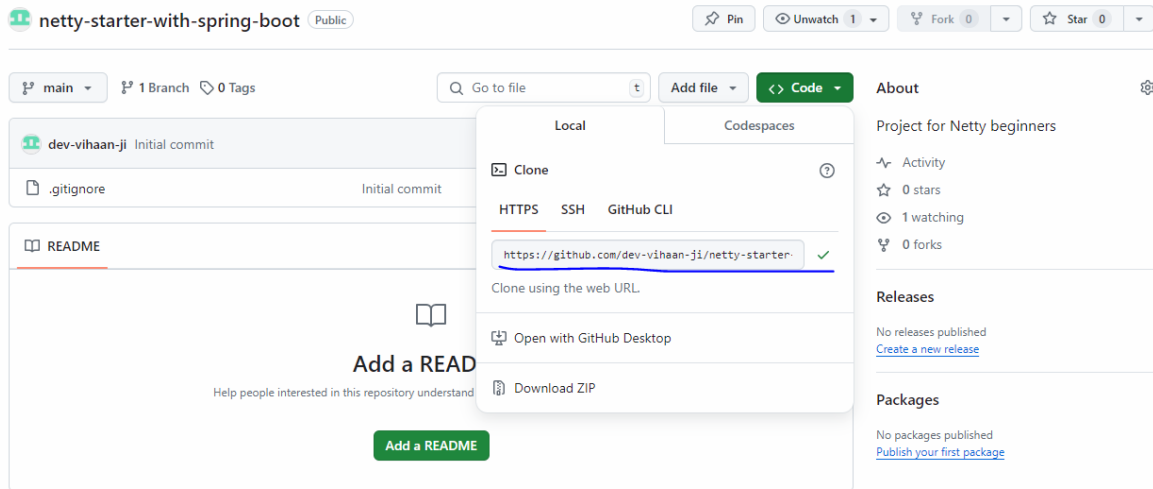
jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot
$ git init
Initialized empty Git repository in D:/jimin/java-project/netty-starter-with-spring-boot/.git/

jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ ls -al
total 33
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:43 ./
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:34 ../
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:43 .git/
-rw-r--r-- 1 jimin 197121   395 Jun 27 22:34 .gitignore
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:35 .idea/
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:34 .mvn/
-rw-r--r-- 1 jimin 197121  1112 Jun 27 22:34 HELP.md
-rwxr-xr-x 1 jimin 197121 10666 Jun 27 22:34 mvnw*
-rw-r--r-- 1 jimin 197121  6913 Jun 27 22:34 mvnw.cmd
-rw-r--r-- 1 jimin 197121  1766 Jun 27 22:34 pom.xml
drwxr-xr-x 1 jimin 197121    0 Jun 27 22:34 src/

jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ .....
```

3.2 .git 파일 생성 후(git remote)

- 파일이 생성됐다면 아래와 같이 원격 GitHub Repository 주소를 복사합니다.



```
jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ git remote -v

jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ git remote add origin https://github.com/dev-vihaan-ji/netty-starter-with-spring-boot.git

jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ git remote -v
origin https://github.com/dev-vihaan-ji/netty-starter-with-spring-boot.git (fetch)
origin https://github.com/dev-vihaan-ji/netty-starter-with-spring-boot.git (push)

jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ |
```

- 그리고 아래와 같이 내 원격 저장소에 연결을 할 수 있습니다. (`git remote add origin` 저장소)
- `git remote -v` 현재 프로젝트에 설정된 원격 저장소의 URL을 확인합니다.

3.3 원격 저장소에서 처음에 파일 가져오기 (git pull)

```
jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 1.86 KiB | 158.00 KiB/s, done.
From https://github.com/dev-vihaan-ji/netty-starter-with-spring-boot
* branch main -> FETCH_HEAD
* [new branch] main -> origin/main
```

- git에 처음으로 연동할 때, 원격 저장소에 `.gitignore`, `README.MD` 파일이 이미 존재하는 경우가 많습니다.

- 이 경우 먼저 `git pull` 브랜치명 명령어를 사용하여 원격 저장소의 파일들을 로컬 저장소로 가져와야 합니다.
- 만약 원격 저장소와 로컬 저장소의 내용이 일치하지 않는 경우, `git pull` 명령어를 실행하면 충돌이 발생할 수 있습니다.
- 이 경우 충돌을 해결한 후 `git add`, `git commit`, `git push` 등의 작업을 수행할 수 있습니다.
- 이렇게 하면 로컬 저장소와 원격 저장소의 내용이 일치하게 되어 이후 `git add`, `git commit`, `git push` 등의 작업을 수행할 수 있습니다.

3.3 원격 저장소에 올릴 파일 Git에 추가 (git add)

```
jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.mvn/wrapper/maven-wrapper.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mvnw', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mvnw.cmd', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'pom.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/java/com/dev/vihaan/netty-starter-with-spring-boot/NettyStarterWithSpringBootApplication.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/resources/application.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/test/java/com/dev/vihaan/netty-starter-with-spring-boot/NettyStarterWithSpringBootApplicationTests.java', LF will be replaced by CRLF the next time Git touches it
```

- `(git add .)` / `(git add *)` → 전체 파일 추가
- `(git add 파일 명)` → 선택한 파일만 추가
- 새로 생성된 파일 또는 작업 디렉토리에서 변경된 파일을 **Git의 Staging Area**에 추가하는 명령어입니다.
- 아래와 같이 warning 로그가 뜨는 이유는 운영체제가 다르기 줄바꿈 문자 특성이 다르기 때문입니다.
- Window OS에서는 줄바꿈 문자 CRLF (4 byte 사용)
- Unix OS에서는 줄바꿈 문자 LF (2 byte 사용)
- Git은 이 차이를 자동으로 처리하려고 시도할 때 이런 경고 메시지가 나타나는 것이며, 실제 코드나 파일에는 영향을 미치지 않습니다.

3.4 Staging Area에 추가된 변경 사항들을 커밋 (git commit)

```
jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ git commit -m "first commit 2024-06-27 vihaan ji"
[main 41a926a] first commit 2024-06-27 vihaan ji
8 files changed, 554 insertions(+)
create mode 100644 .gitignore
create mode 100644 .mvn/wrapper/maven-wrapper.properties
create mode 100644 mvnw
create mode 100644 mvnw.cmd
create mode 100644 pom.xml
create mode 100644 src/main/java/com/dev/vihaan/netty/starter/with/spring_boot/NettyStarterWithSpringBootApplication.java
create mode 100644 src/main/resources/application.properties
create mode 100644 src/test/java/com/dev/vihaan/netty/starter/with/spring_boot/NettyStarterWithSpringBootApplicationTests.java
```

- Staging Area에 추가된 변경 사항들을 하나의 커밋으로 기록하는 명령어입니다.

- 커밋 메시지를 작성하여 변경 사항에 대한 설명을 남길 수 있습니다.
- 커밋은 로컬 저장소에 기록됩니다.

3.5 원격 저장소에 업로드 (git push)

```
jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$ git push origin main
Enumerating objects: 33, done.
Counting objects: 100% (33/33), done.
Delta compression using up to 8 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (32/32), 8.93 KiB | 2.23 MiB/s, done.
Total 32 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/dev-vihaan-ji/netty-starter-with-spring-boot.git
b91b6f4..41a926a main -> main
```

- 로컬 저장소에 있는 커밋들을 원격 저장소(예: GitHub)에 업로드하는 명령어입니다.
- 로컬 저장소의 커밋 내역을 원격 저장소에 반영합니다.
- 3.5.1 에러 참조

3.5.1 git push 에러 (자주 겪는 문제라 참조하라고 넣었습니다.)

```
$ git push origin main
To https://github.com/dev-vihaan-ji/netty-starter-with-spring-boot.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/dev-vihaan-ji/netty-starter-with-spring-boot.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

jimin@DESKTOP-2AFR61T MINGW64 /d/jimin/java-project/netty-starter-with-spring-boot (main)
$
```

- 이 오류 메시지는 원격 저장소에 로컬 저장소와 다른 커밋이 존재하여 푸시가 실패한 것을 의미합니다.
- 이러한 경우 일반적으로 아래와 같은 해결책을 사용합니다:
- 원격 저장소 변경 내용 가져오기: `git pull 브랜치`
- 명령어를 사용하여 원격 저장소의 변경 내용을 로컬로 가져옵니다. 이 과정에서 충돌이 발생할 수 있으므로, 충돌을 해결하고 커밋합니다.
- 그래서 보통 git에 처음 연동할 때 원격 저장소에 `.gitignore`, `README.MD` 파일이 존재하는 경우에는 먼저 pull을 받고 작업을 해야 합니다.

3.6 원격 저장소 확인 성공

- 아래와 같이 정상적으로 GitHub Repository에 업로드한 것을 확인할 수 있습니다.

The screenshot shows the GitHub repository page for 'dev-vihaan-ji / netty-starter-with-spring-boot'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows a first commit on 2024-06-27 by vihaan ji. The file list includes .mvn/wrapper, src, .gitignore, mvnw, mvnw.cmd, and pom.xml, all committed on 2024-06-27. The README section is currently empty, with a prompt to 'Add a README'.

dev-vihaan-ji / netty-starter-with-spring-boot

Issues Pull requests Actions Projects Wiki Security Insights Settings

netty-starter-with-spring-boot Public Pin Unwatch 1

main 1 Branch 0 Tags Go to file Add file Code

dev-vihaan-ji first commit 2024-06-27 vihaan ji 41a926a · 9 minutes ago 3 Commits

.mvn/wrapper	first commit 2024-06-27 vihaan ji	9 minutes ago
src	first commit 2024-06-27 vihaan ji	9 minutes ago
.gitignore	first commit 2024-06-27 vihaan ji	9 minutes ago
mvnw	first commit 2024-06-27 vihaan ji	9 minutes ago
mvnw.cmd	first commit 2024-06-27 vihaan ji	9 minutes ago
pom.xml	first commit 2024-06-27 vihaan ji	9 minutes ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README