



FFmpeg

1. FFmpeg 설치

FFmpeg는 오디오 및 비디오 파일을 처리하는 강한 오픈소스.
사용하기 위해서 서버에 FFmpeg를 설치해야 함.

설치

- **Ubuntu:**
 - `sudo apt-get update`
 - `sudo apt-get install ffmpeg`
- **Windows**
 - 바이너리 파일 다운로드
 - 압축을 풀고, `bin` 디렉토리 경로를 시스템 환경 변수 `Path`에 추가

2. 프로젝트에 FFmpeg 라이브러리 추가

```
implementation 'net.bramp.ffmpeg:ffmpeg:0.6.2'
```

3. FFmpeg 경로 설정

- **application.properties**

```
ffmpeg.path=/path/to/ffmpeg  
ffprobe.path=/path/to/ffprobe
```

- **Java 코드에서 경로 주입**

```
@Value("${ffmpeg.path}")  
private String ffmpegPath;  
  
@Value("${ffprobe.path}")  
private String ffprobePath;
```

4. FFmpeg 초기화

FFmpeg와 FFprobe 객체를 초기화

```
@PostConstruct  
public void init() {  
    try {  
        FFmpeg ffmpeg = new FFmpeg(ffmpegPath);  
        FFprobe ffprobe = new FFprobe(ffprobePath);  
        System.out.println("FFmpeg 및 FFprobe 초기화 완료");  
    } catch (IOException e) {  
        throw new RuntimeException("FFmpeg 초기화 실패", e);  
    }  
}
```

5. FFmpeg 작업 수행

음성 파일의 잡음 제거

- anlmdn 필터 : FFmpeg에서 제공하는 간단한 잡음 제거 필터
 - `ffmpeg -i input.wav -af anlmdn=s=10 output.wav`
- arnndn 필터 : FFmpeg의 RNNoise 기반 딥러닝 잡음 제거 필터로, 더 정교한 결과를 제공
 - `ffmpeg -i input.wav -af arnndn=m=model.rnnn output.wav`
- 고/저주파 대역 필터링
 - `ffmpeg -i input.wav -af "highpass=f=300, lowpass=f=3000" output.wav`
 - `highpass=f=300` : 300Hz 이하의 저주파를 제거.
 - `lowpass=f=3000` : 3000Hz 이상의 고주파를 제거

두 파일의 병

- amix 필터 (믹싱) : 두 음성을 동일한 트랙으로 혼
 - `ffmpeg -i input1.wav -i input2.wav -filter_complex amix=inputs=2:duration=longest output.wav`
 - `inputs=2` : 입력 파일 개수.
 - `duration=longest` : 가장 긴 입력 파일의 길이에 맞춰 출력 길이를 설정
- amerge 필터 : 두 음성을 각각 다른 채널(예: 좌/우)로 병합하여 스테레오 트랙 생성
 - `ffmpeg -i input1.wav -i input2.wav -filter_complex "[0:a][1:a]amerge=inputs=2[a]" -map "[a]" output.wav`
 - 좌/우 채널로 분리된 스테레오 오디오가 생성됨

6. 구현 예제

```
import java.io.IOException;

public class AudioProcessor {
    public static void main(String[] args) {
        try {
            // 잡음 제거 명령어 실행
            String denoiseCommand = "ffmpeg -i input.wav -af anlmdn=s=10 outp
            Process denoiseProcess = Runtime.getRuntime().exec(denoiseComma
            denoiseProcess.waitFor();
```

```

// 두 음성 파일 병합 명령어 실행
String mergeCommand = "ffmpeg -i input1.wav -i input2.wav -filter_co
Process mergeProcess = Runtime.getRuntime().exec(mergeCommand);
mergeProcess.waitFor();

System.out.println("작업 완료");
} catch (IOException | InterruptedException e) {
    e.printStackTrace();
}
}
}

```

메모

- 커맨드 단위로 작업이 되는 듯 함.
- 임시 파일을 저장하고, 저장된 임시 파일로 작업을 한 후 파일을 기록해야함.
- 임시 파일들이 쌓이지 않도록 삭제 체크를 잘 해주어야 함.