Trackplay

- 서버에 업로드된 트랙 파일을 실시간 http 스트리밍하는 데에는 `ExoPlayer` 적합
- `Jetpack Media3` 라이브러리를 사용하면 exoplayer와 ui까지 함께 가져올 수 있음
- 저지연 최적화는 되어있지 않지만 mp3, acc 등으로 최적화된 음원 스트리밍 가능
- 간단한 이퀄라이저 설정 등도 추가 가능
  - 사용자가 import 전에 테스트해볼 수 있지 않을까?
- `media3 session`을 이용하면 백그라운드 재생 구현 가능
- 테스트용 음원 링크

```
https://whistlehub.s3.ap-northeast-2.amazonaws.com/%EA%B5%AD%EC
```

🚀 Oboe vs ExoPlayer (서버 음원 재생용)

기능	Oboe	ExoPlayer
MP3, AAC 스트리밍 지원	❌ (직접 구현 필요)	✅ (내장 지원)
HTTP 스트리밍	❌ (파일 직접 다운로드 후 처리 필요)	✅ (HLS, DASH 지원)
저지연(Local File 재생)	✅	❌
오디오 효과 (이퀄라이저 등)	❌ (Superpowered 필요)	✅ (기본 제공)
간편한 사용 (Kotlin)	❌ (JNI 필요)	✅ (Kotlin 코드만으로 가능)

1️⃣ Gradle 설정 (`Media3` 라이브러리 추가)

```
dependencies {
    implementation(libs.androidx.media3.exoplayer)
    implementation(libs.androidx.media3.exoplayer.dash)
    implementation(libs.androidx.media3.ui.compose)
    implementation(libs.androidx.media3.session)
}
```

2️⃣ AndroidManifest.xml (Foreground Service 등록)

```
<serviceandroid:name=".AudioPlayerService"
    android:foregroundServiceType="mediaPlayback"
    android:exported="false">
    <intent-filter>
        <action android:name="android.media.browse.MediaBrowserService"
/>
    </intent-filter>
</service>

<!-- Foreground Service 권한 추가 -->
<uses-permission android:name="android.permission.FOREGROUND_SER
VICE_MEDIA_PLAYBACK"/>
```

✔️ **Foreground Service 등록**

✔️ **Android 9+에서 필요한 Foreground Service 권한 추가**

3️⃣ AudioPlayerService (백그라운드 재생)

```
class AudioPlayerService : MediaSessionService() {
    private lateinit var mediaSession: MediaSession
    private lateinit var player: ExoPlayer

    override fun onCreate() {
        super.onCreate()
        player = ExoPlayer.Builder(this).build()
        mediaSession = MediaSession.Builder(this, player).build()

        val mediaItem = MediaItem.fromUri("https://your-server.com/audio.mp
3")
        player.setMediaItem(mediaItem)
        player.prepare()
    }

    override fun onDestroy() {
        mediaSession.release()
        player.release()
        super.onDestroy()
```

```
    }

    override fun onGetSession(controllerInfo: MediaSession.ControllerInfo):
MediaSession {
        return mediaSession
    }
}
```

✔️ **MediaSession + ExoPlayer 통합**

✔️ **앱 종료 후에도 음악 유지**

4️⃣ Foreground Service 시작하는 함수

```
fun startAudioService(context: Context) {
    val intent = Intent(context, AudioPlayerService::class.java)
    ContextCompat.startForegroundService(context, intent)
}
```

✔️ **앱이 실행될 때 백그라운드 서비스 시작**

5️⃣ Jetpack Compose에서 ExoPlayer + Media3 UI 사용

```
@Composable
fun AudioPlayerScreen(context: Context) {
    val player = remember {
        ExoPlayer.Builder(context).build().apply {
            val mediaItem = MediaItem.fromUri("https://your-server.com/audio.
mp3")
            setMediaItem(mediaItem)
            prepare()
        }
    }

    val lifecycleOwner = LocalLifecycleOwner.current

    AndroidView(
        factory = { ctx →
            StyledPlayerView(ctx).apply {
```

```
                this.player = player
                useController = true // 미디어 컨트롤러 활성화
            }
        },
        modifier = Modifier.fillMaxWidth().height(300.dp)
    )

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Button(onClick = {
            startAudioService(context) // Foreground Service 실행
            player.play()
        }) {
            Text("재생")
        }
        Spacer(modifier = Modifier.height(8.dp))
        Button(onClick = { player.pause() }) {
            Text("일시정지")
        }
        Spacer(modifier = Modifier.height(8.dp))
        Button(onClick = { player.stop() }) {
            Text("정지")
        }
    }

    DisposableEffect(lifecycleOwner) {
        onDispose {
            player.release() // 메모리 해제
        }
    }
}
```

✔ **Media3의 `StyledPlayerView`를 Jetpack Compose에서 사용**

✔ **백그라운드 서비스 실행 버튼 추가**

✔ **Lifecycle 관리 (재생 종료 시 `player.release()` 호출)**

플로우

- 트랙을 누르면 현재 재생목록에 추가되면서 재생이 된다.
- 재생중인 트랙은 일시정지할 수 있다.
- 플레이어는 네비게이션 바와 함께 따라다닌다.
  (일부 페이지-daw등 제외하고는 플레이어가 사라지지 않는다. ⇒ 일시정지 상태로 보임)