

[API 참고서 >](#) [설정 >](#)

logger

logger 옵션은 컴파일 중 React 컴파일러 이벤트에 대한 커스텀 로깅을 제공합니다.

```
{  
  logger: {  
    logEvent(filename, event) {  
      console.log(`[Compiler] ${event.kind}: ${filename}`);  
    }  
  }  
}
```

- [레퍼런스](#)
 - [logger](#)
- [사용법](#)
 - [기본 로깅](#)
 - [상세 오류 로깅](#)

레퍼런스

logger

컴파일러 동작을 추적하고 문제를 디버깅하기 위한 커스텀 로깅을 설정합니다.

타입

```
{  
  logEvent: (filename: string | null, event: LoggerEvent) => void;  
}
```

```
} | null
```

기본값

null

메서드

- `logEvent`: 각 컴파일러 이벤트에 대해 파일명, 이벤트 세부 정보와 함께 호출됩니다.

이벤트 타입

- `CompileSuccess`: 함수가 성공적으로 컴파일됨.
- `CompileError`: 오류로 인해 함수가 건너뛰어짐.
- `CompileDiagnostic`: 치명적이지 않은 진단 정보.
- `CompileSkip`: 다른 이유로 함수가 건너뛰어짐.
- `PipelineError`: 예기치 않은 컴파일 오류.
- `Timing`: 성능 타이밍 정보.

주의 사항

- 이벤트 구조는 버전 간에 변경될 수 있습니다.
- 대규모 코드베이스는 많은 로그 항목을 생성합니다.

사용법

기본 로깅

컴파일 성공과 실패를 추적합니다.

```
{
  logger: {
    logEvent(filename, event) {
      switch (event.kind) {
        case 'CompileSuccess': {
          console.log(`✓ Compiled: ${filename}`);
          break;
        }
      }
    }
  }
}
```

```
        }

        case 'CompileError': {
            console.log(`✖ Skipped: ${filename}`);
            break;
        }
        default: {}
    }
}

}

}
```

상세 오류 로깅

컴파일 실패에 대한 구체적인 정보를 확인합니다.

```
{
logger: {

    logEvent(filename, event) {
        if (event.kind === 'CompileError') {
            console.error(`\nCompilation failed: ${filename}`);
            console.error(`Reason: ${event.detail.reason}`);

            if (event.detail.description) {
                console.error(`Details: ${event.detail.description}`);
            }

            if (event.detail.loc) {
                const { line, column } = event.detail.loc.start;
                console.error(`Location: Line ${line}, Column ${column}`);
            }
        }

        if (event.detail.suggestions) {
            console.error('Suggestions:', event.detail.suggestions);
        }
    }
}
}
```

이전

gating

다음

panicThreshold

 Meta Open Source

Copyright © Meta Platforms, Inc

uwu?

React 학습하기

빠르게 시작하기

설치하기

UI 표현하기

상호작용성 더하기

State 관리하기

탈출구

API 참고서

React APIs

React DOM APIs

커뮤니티

행동 강령

팀 소개

문서 기여자

감사의 말

더 보기

블로그

React Native

개인 정보 보호

약관

