



gating

`gating` 옵션은 조건부 컴파일을 활성화하여 런타임에 최적화된 코드가 사용되는 시점을 제어할 수 있게 합니다.

```
{
  gating: {
    source: 'my-feature-flags',
    importSpecifierName: 'shouldUseCompiler'
  }
}
```

- [레퍼런스](#)

- [gating](#)

- [사용법](#)

- [기본 기능 플래그 설정](#)

- [문제 해결](#)

- [기능 플래그가 작동하지 않는 경우](#)
- [Import 오류](#)

레퍼런스

`gating`

컴파일된 함수에 대한 런타임 기능 플래그 `Gating`을 설정합니다.

타입

```
{
  source: string;
  importSpecifierName: string;
} | null
```

기본값

null

프로퍼티

- **source**: 기능 플래그를 가져올 모듈 경로.
- **importSpecifierName**: import 해서 사용하려는 내보낸^{Export} 함수의 이름.

주의 사항

- Gating 함수는 반드시 boolean을 반환해야 합니다.
- 컴파일된 버전과 원본 버전 모두 번들 크기를 증가시킵니다.
- import 는 컴파일된 함수가 있는 모든 파일에 추가됩니다.

사용법

기본 기능 플래그 설정

1. 기능 플래그 모듈을 생성합니다.

```
// src/utils/feature-flags.js
export function shouldUseCompiler() {
  // your logic here
  return getFeatureFlag('react-compiler-enabled');
}
```

2. 컴파일러를 설정합니다.

```
{
```

```
gating: {
  source: './src/utils/feature-flags',
  importSpecifierName: 'shouldUseCompiler'
}
}
```

3. 컴파일러가 게이트된 코드를 생성합니다.

```
// Input
function Button(props) {
  return <button>{props.label}</button>;
}

// Output (simplified)
import { shouldUseCompiler } from './src/utils/feature-flags';

const Button = shouldUseCompiler()
  ? function Button_optimized(props) { /* compiled version */ }
  : function Button_original(props) { /* original version */ };
```

Gating 함수는 모듈 시간에 한 번만 평가되므로, JS 번들이 파싱되고 평가된 후에는 선택된 컴포넌트가 브라우저 세션이 끝날 때까지 정적으로 유지됩니다.

문제 해결

기능 플래그가 작동하지 않는 경우

플래그 모듈이 올바른 함수를 내보내는지 확인하세요.

```
// ❌ 잘못된 예: Default export
export default function shouldUseCompiler() {
  return true;
}

// ✅ 올바른 예: importSpecifierName과 일치하는 Named export
export function shouldUseCompiler() {
```

```
return true;
}
```

Import 오류

source 의 경로가 올바른지 확인하세요.

```
// ❌ 잘못된 예: `babel.config.js`에 상대적인 경로
{
  source: './src/flags',
  importSpecifierName: 'flag'
}

// ✅ 올바른 예: 모듈 해석 경로
{
  source: '@myapp/feature-flags',
  importSpecifierName: 'flag'
}

// ✅ 올바른 예: 프로젝트 루트로부터의 절대 경로
{
  source: './src/utils/flags',
  importSpecifierName: 'flag'
}
```

< [이전](#)
[compilationMode](#)

[다음](#) >
[logger](#)

uwu?

설치하기

React DOM APIs

UI 표현하기

상호작용성 더하기

State 관리하기

탈출구

커뮤니티

행동 강령

팀 소개

문서 기여자

감사의 말

더 보기

블로그

React Native

개인 정보 보호

약관

