



프로덕션 배포

개발 vs. 프로덕션

개발 중에는 Vue가 개발 경험을 향상시키기 위한 여러 기능을 제공합니다:

일반적인 오류 및 함정에 대한 경고

Props / 이벤트 유효성 검사

반응성 디버깅 툐

Devtools 통합

하지만 이러한 기능들은 프로덕션 환경에서는 쓸모가 없습니다. 일부 경고 체크는 소량의 성능 오버헤드를 유발할 수도 있습니다. 프로덕션에 배포할 때는 사용하지 않는 개발 전용 코드 분기 를 모두 제거하여 페이지 크기를 줄이고 성능을 향상시켜야 합니다.

빌드 도구 없이

CDN이나 자체 호스팅 스크립트에서 Vue를 로드하여 빌드 도구 없이 사용하는 경우, 프로덕션에 배포할 때는 반드시 프로덕션 빌드(파일명이 `.prod.js`로 끝나는 dist 파일)를 사용해야 합니다. 프로덕션 빌드는 모든 개발 전용 코드 분기가 제거된 상태로 미리 압축되어 있습니다.

글로벌 빌드(`Vue` 전역을 통해 접근) 사용 시: `vue.global.prod.js` 를 사용하세요.

ESM 빌드(네이티브 ESM import를 통해 접근) 사용 시: `vue.esm-browser.prod.js` 를 사용하세요.

자세한 내용은 `dist` 파일 가이드를 참고하세요.

빌드 도구 사용 시



맞게 사전 구성되어 있습니다.

커스텀 설정을 사용하는 경우, 다음을 반드시 확인하세요:

1. vue 가 vue.runtime.esm-bundler.js 로 resolve됩니다.
2. 컴파일 타임 기능 플래그가 올바르게 설정되어 있습니다.
3. 빌드 시 process.env.NODE_ENV 가 "production" 으로 대체됩니다.

추가 참고 자료:

[Vite 프로덕션 빌드 가이드](#)

[Vite 배포 가이드](#)

[Vue CLI 배포 가이드](#)

런타임 오류 추적

앱 레벨 오류 핸들러를 사용하여 오류를 추적 서비스에 보고할 수 있습니다:

```
import { createApp } from 'vue' js

const app = createApp(...)

app.config.errorHandler = (err, instance, info) => {
  // 오류를 추적 서비스에 보고
}
```

Sentry 및 Bugsnag과 같은 서비스는 Vue를 위한 공식 통합도 제공합니다.

[GitHub에서 이 페이지 편집](#)

< Previous

서버 사이드 렌더링 (SSR)

Next >

성능