

프로덕션 배포

개발(Development) vs. 프로덕션(Production)

개발하는 동안 Vue는 개발 경험을 개선하기 위해 다음과 같은 여러 기능을 제공합니다:

- 일반적인 에러 및 위험에 대한 경고
- props/이벤트 유효성 검사
- 반응성 디버깅
- Devtools 통합

그러나 이러한 기능은 프로덕션에선 필요 없습니다. 일부 경고 확인은 약간의 성능 오버헤드를 유발할 수도 있습니다. 프로덕션에 배포할 때 더 작은 페이로드 크기와 더 나은 성능을 위해 사용되지 않는 개발 전용 코드 분기를 모두 삭제해야 합니다.

빌드 툴 없이

CDN 또는 자체 호스팅 스크립트에서 로드하여 빌드 도구 없이 Vue를 사용하는 경우, 프로덕션에 배포할 때 프로덕션 빌드(`.prod.js` 로 끝나는 `dist` 파일)를 사용해야 합니다. 프로덕션 빌드는 모든 개발 전용 코드 분기가 제거된 상태로 미리 최소화됩니다.

- 전역 빌드를 사용하는 경우(`Vue` 전역을 통해 접근): `vue.global.prod.js` 를 사용합니다.
- ESM 빌드를 사용하는 경우(네이티브 ESM 가져오기를 통해 접근): `vue.esm-browser.prod.js` 를 사용합니다.

자세한 내용은 `dist` 파일 가이드를 참고하세요.

빌드 툴 사용

`create-vue` (Vite 기반) 또는 `Vue CLI`(webpack 기반)를 통해 스캐폴딩된 프로젝트는 프로덕션 빌드를 위해 사전 구성됩니다.

커스텀 설정을 사용하는 경우 다음을 확인하십시오:

- `vue` 는 `vue.runtime.esm-bundler.js` 로 해결(resolves).
- 컴파일 시간 기능 플래그가 올바르게 구성됨.
- `process.env.NODE_ENV` 는 빌드 중에 `"production"` 으로 대체.

참고:

- Vite 프로덕션 빌드 가이드
- Vite 배포 가이드
- Vue CLI 배포 가이드

런타임 에러 추적

앱 수준 에러 핸들러를 사용하여 에러 보고를 추적 제공할 수 있습니다:

```
import { createApp } from 'vue'  
  
const app = createApp(...)
```

js

```
app.config.errorHandler = (err, instance, info) => {  
  // 에러 보고를 추적 제공  
}
```

Sentry 및 Bugsnag와 같은 서비스도 Vue에 대한 공식 통합을 제공합니다.