

# LU16c - Bundling, Minify & Deploy

Bevor eine Vue.js-App veröffentlicht werden kann, muss sie **gebündelt** werden. Dieser Schritt optimiert alle Dateien für den Produktionseinsatz und erstellt einen `dist`-Ordner, der auf einem Server bereitgestellt werden kann.

## npm run build

Der Befehl `npm run build` startet den Vite-Build-Prozess.

```
npm run build
```

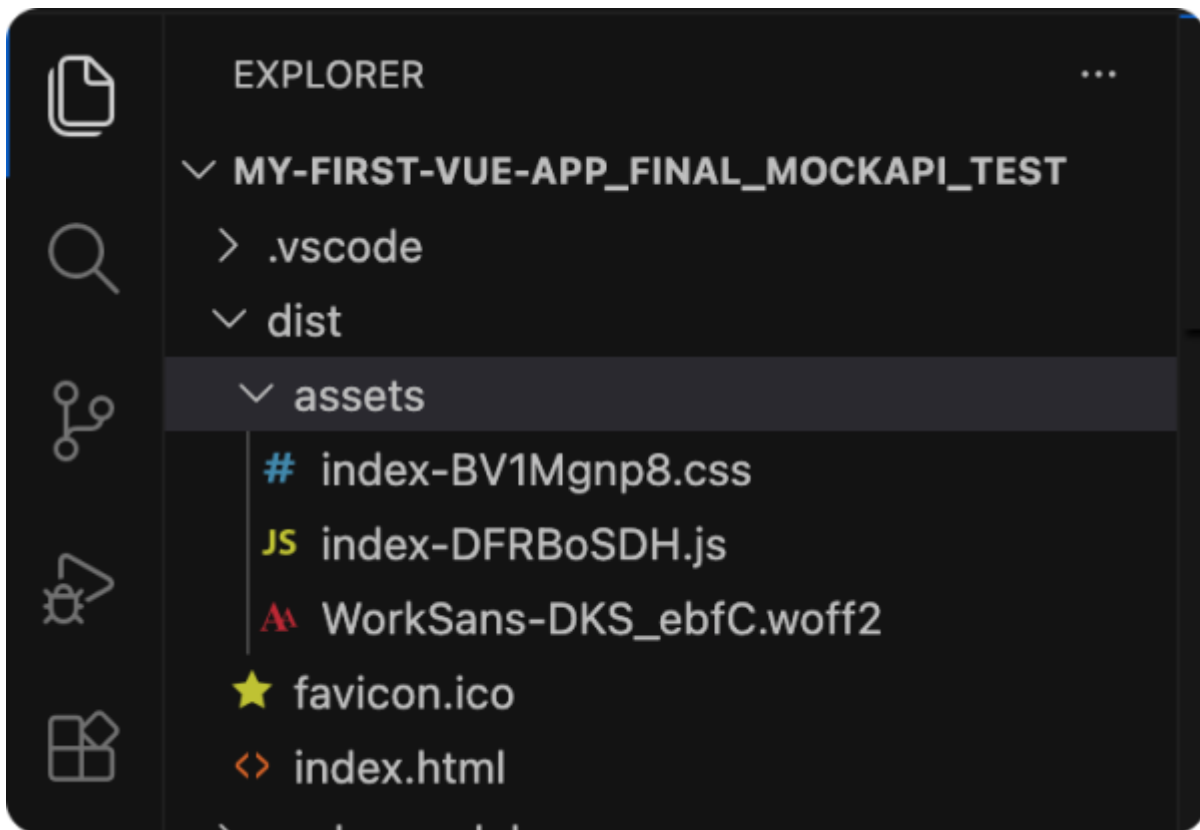
```
291_WebUI/16_LU/my-first-vue-app_final_mockapi_test via ● v24.12.0 using ☁ temp-config/
→ npm run build

> my-first-vue-app@0.0.0 build
> vite build

vite v7.3.6 building client environment for production...
✓ 16 modules transformed.
dist/index.html                0.43 kB | gzip: 0.28 kB
dist/assets/WorkSans-DKS_ebfC.woff2 131.31 kB
dist/assets/index-BV1Mgnp8.css   5.53 kB | gzip: 2.18 kB
dist/assets/index-DFRBoSDH.js   63.56 kB | gzip: 25.39 kB
✓ built in 297ms
```

Platzhalter: Screenshot Terminal mit Build-Output - Dateigrößen, Chunk-Namen, Dauer

Nach dem Build liegt ein neuer Ordner `dist/` im Projekt:



Screenshot VS Code Explorer mit dist-Ordner: index.html, assets/-Unterordner mit .js und .css Dateien

Dieser dist/-Ordner enthält alles, was für den Betrieb der App nötig ist – kein Node.js, kein Vite, keine Vue-Komponenten.


## Minification

**Minification** bedeutet: überflüssige Zeichen aus dem Code entfernen, ohne die Funktion zu verändern.

Vorher (Entwicklung)	Nachher (Production)
Zeilenumbrüche, Einrückungen	Alles in einer Zeile
Lange Variablennamen	Kurze Namen (a, b, fn)
Kommentare	Entfernt

inader - links normal, rechts minifiziert

**Ziel:** Kleinere Dateigrösse → kürzere Ladezeit für die Nutzerin / den Nutzer.



Vite minifiziert automatisch bei `npm run build`. Sie müssen nichts zusätzlich konfigurieren.

## Chunking & Tree Shaking

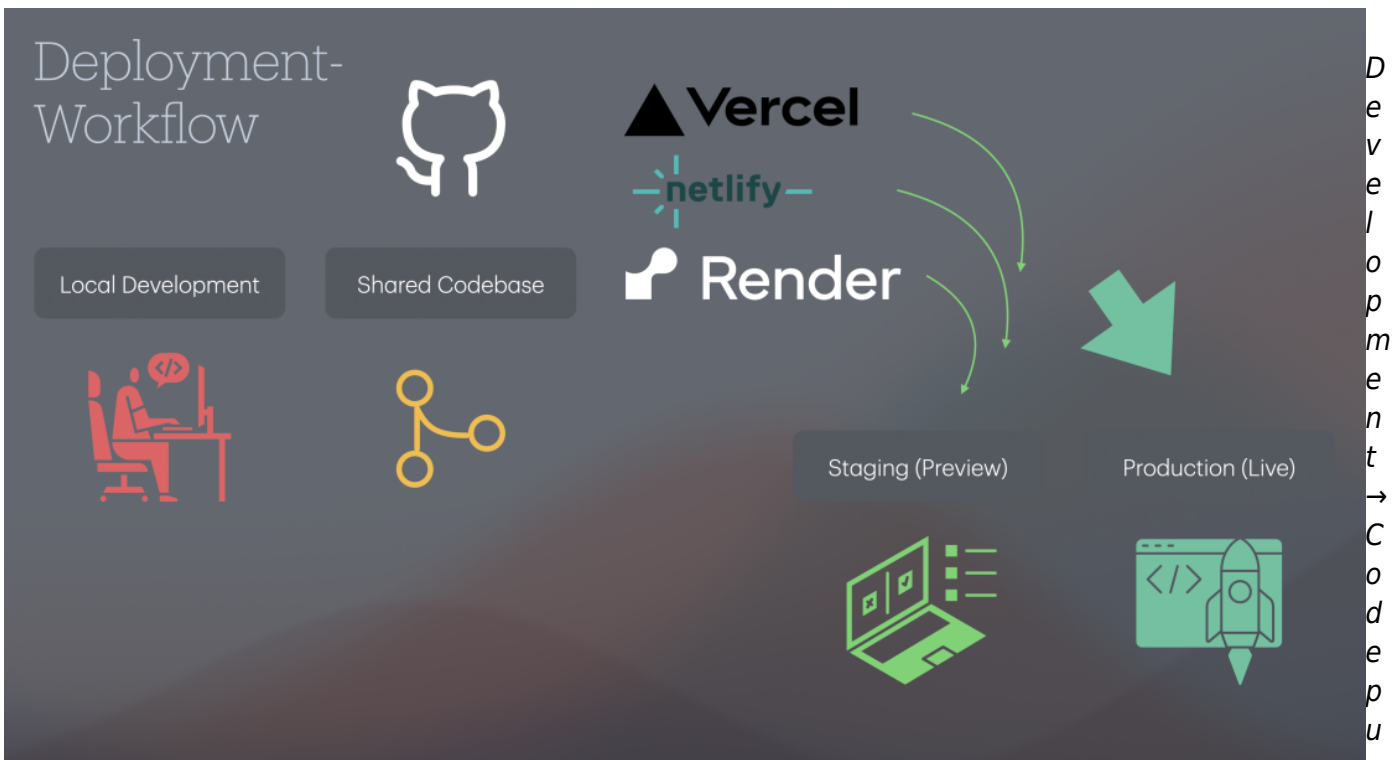
Vite teilt den JavaScript-Code automatisch in **mehrere kleine Dateien (Chunks)** auf, anstatt alles in eine grosse Datei zu schreiben.

**Chunking:** Der Browser lädt nur den Code, den er gerade braucht - nicht alles auf einmal.

**Tree Shaking:** Code, der im Projekt importiert aber nie verwendet wird, wird automatisch aus dem Build entfernt.

## Deploy: Mit Git (Vercel, Netlify, Render)

Die meisten modernen Hosting-Plattformen verbinden sich mit einem GitHub-Repository und deployen automatisch, sobald neuer Code gepusht wird.



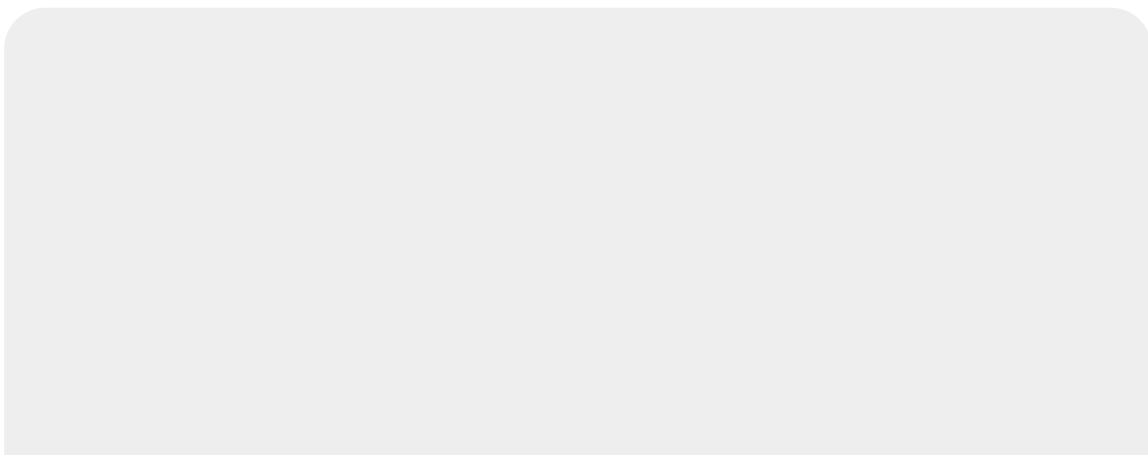
h zu GitHub → Vercel/Netlify/Render → Preview → Production (Live)

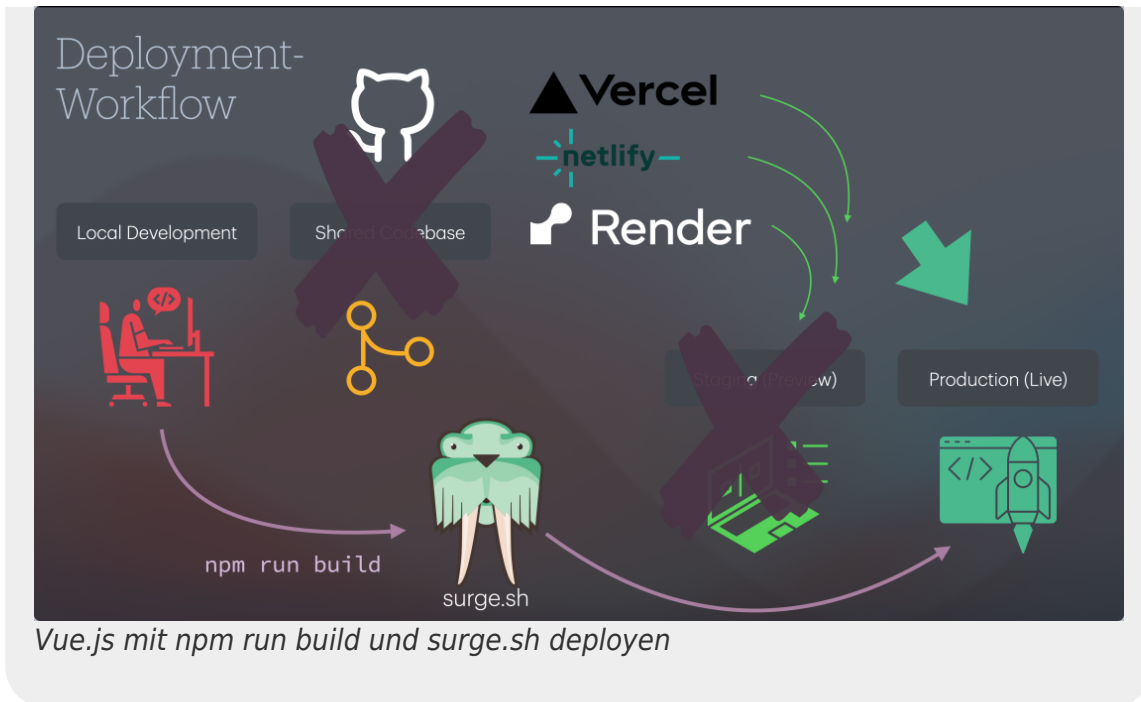
Anbieter	Kostenlos	Besonderheit
Vercel	☐ (Hobby-Plan)	Sehr schnell, optimiert für Vue/React/Next.js
Netlify	☐ (Starter-Plan)	Netlify Drop für direktes Hochladen möglich
Render	☐ (mit Einschränkungen)	Auch für Backend-Services geeignet

**Voraussetzung:** Ein GitHub-Account und ein Repository mit dem Projektcode. Diese Plattformen eignen sich ideal für Teams, die mit Git arbeiten.

## Deploy: Ohne Git (Surge.sh & Netlify Drop)

Für den Unterricht – ohne Git-Kenntnisse – gibt es zwei einfachere Wege:





### Option A: Surge.sh (Terminal-Weg)

```
npm run build
cd dist
npx surge
```

Surge fragt beim ersten Start nach E-Mail und Passwort (kostenlose Registrierung direkt im Terminal).  
Anschließend wählen Sie eine Domain:

```
domain: vorname-faq.surge.sh
```



Screenshot Terminal – surge-Befehl, Registrierung, gewählte Domain, grüne Erfolgsanzeige

□ Die App ist live unter <https://vorname-faq.surge.sh>

## Option B: Netlify Drop (Drag-and-Drop)

1. npm run build ausführen
2. [app.netlify.com/drop](https://app.netlify.com/drop) im Browser öffnen

3. Den `dist/`-Ordner per Drag-and-Drop ins Browserfenster ziehen
4. Fertig - Netlify erstellt sofort eine öffentliche URL

**Hinweis:** Ohne Konto ist die Netlify-Drop-URL nur ca. 1 Stunde aktiv. Mit kostenlosem Konto (keine Kreditkarte nötig) bleibt sie dauerhaft erreichbar.

## Vergleich: Deploy-Optionen

	Surge.sh	Netlify Drop	Vercel / Netlify / Render
<b>Git nötig?</b>	Nein	Nein	Ja
<b>Terminal nötig?</b>	Ja	Nein	Nein
<b>Aufwand</b>	Gering	Sehr gering	Mittel (Einrichtung)
<b>Geeignet für</b>	Unterricht, Demos	Unterricht, schnelle Tests	Echte Projekte
<b>Automatisches Re-Deploy</b>	Nein	Nein	Ja (bei Git-Push)
<b>Gratis</b>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> (mit Limits)

## Zusammenfassung: Der komplette Ablauf

```
# 1. App für Production optimieren
npm run build

# 2. In den dist-Ordner wechseln
cd dist

# 3. Deployen mit Surge
npx surge
```

Nach diesen drei Befehlen ist die Vue.js-App unter einer echten, öffentlichen URL erreichbar.

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
[https://wiki.bzz.ch/de/modul/m291/learningunits/lu16/theorie/c\\_deploy?rev=1782683343](https://wiki.bzz.ch/de/modul/m291/learningunits/lu16/theorie/c_deploy?rev=1782683343)

Last update: **2026/06/28 23:49**

