Parcel Zero-config bundler

START HERE

`npm install -D parcel`
Add Parcel to a project

`npx parcel --port 3000`
Pick the dev port

`npx parcel src/index.html`
Dev server with HMR

`npx parcel build --dist-dir out`
Change output folder

`npx parcel build src/index.html`
Optimized production build

`npx parcel build --public-url /app/`
Deploy under a sub-path

`npx parcel watch src/index.js`
Rebuild only, no server

`npx parcel build --no-cache`
Build without the cache

Entry index.html or source field → `parcel` → **Dev server** localhost:1234, HMR → `parcel build` → **dist/** minified, hashed files → `parcel watch` → **Watch** rebuild, no HTTP server

CONCEPTS

Zero config: point Parcel at an entry (usually an HTML file) and it follows script, style and image references itself. serve is the default command.

The cache lives in .parcel-cache; add it and dist to .gitignore. When output looks stale or odd, delete the cache or pass --no-cache.

DEV SERVER

`parcel` ***<entry>***
Start the dev server (same as parcel serve)

`parcel` **`serve`** ***<entry>***
Explicit form of the dev server

`parcel` **`--port`** ***<port>***
Set the port (default 1234 or $PORT)

`parcel` **`--host`** ***<host>***
Listen on one host instead of all

`parcel` **`--open [browser]`**
Open the entry in a browser

`parcel` **`--https`**
Serve dev and HMR over HTTPS

`parcel` **`--cert`** ***<file>*** **`--key`** ***<file>***
Use your own HTTPS certificate

`parcel` **`--no-hmr`**
Turn off hot reloading

`parcel` **`--lazy`**
Build only bundles the browser requests
Speeds up first start in big apps.

`parcel` **`--hmr-port`** ***<port>***
Run the HMR socket on another port

`parcel` **`--no-autoinstall`**
Do not auto-install missing packages

WATCH

`parcel` **`watch`** ***<entry>***
Rebuild on change, no HTTP server

`parcel` **`watch --no-hmr`**
Watch without the HMR server

`parcel` **`watch --watch-dir`** ***<dir>***
Set the root directory to watch
Useful in monorepos with nested lockfiles.

`parcel` **`watch --watch-for-stdin`**
Exit when stdin closes

BUILD

`parcel` **`build`** ***<entry>***
One production build, then exit

`parcel` **`build 'src/*.html'`**
Build many entries by glob (quote it)

`parcel` **`build`**
Use the source field of package.json

`parcel` **`build --no-optimize`**
Skip minification and optimizations

`parcel` **`build --no-scope-hoist`**
Disable scope hoisting

`parcel` **`build --no-content-hash`**
Keep names free of content hashes

`parcel` **`build --detailed-report [n]`**
List the n largest assets per bundle

`parcel` **`build --no-source-maps`**
Do not emit source maps

OUTPUT AND TARGETS

`--dist-dir` ***<dir>***
Output folder (default dist)

`--public-url` ***<url>***
Prefix for absolute URLs in output

`--target` ***<name>***
Build only the named target
Targets come from the targets key in package.json.

package.json fields

Field	Purpose
`source`	Default entry file(s)
`main`	Library output path (CommonJS)
`module`	Library output path (ES module)
`browserslist`	Browsers to support
`targets`	Per-target dist, publicUrl, options

--dist-dir and --public-url only apply where a target does not set its own distDir or publicUrl.

CACHE AND DIAGNOSTICS

`--no-cache`
Do not read from the cache

`--cache-dir` ***<path>***
Move the cache (default .parcel-cache)

`--log-level` ***<level>***
none, error, warn, info or verbose

`--config` ***<path>***
Use another Parcel config or package

`--reporter` ***<pkg>***
Add a reporter plugin (repeatable)

`--profile`
Record a CPU profile of the build

CACHE AND DIAGNOSTICS (cont.)

`--trace`
Record a trace of the build

`parcel` **`--version`**
Print the Parcel version

CONFIG FILES

Minimal .parcelrc

```
{
  "extends": "@parcel/config-default",
  "transformers": { "*.svg": ["@parcel/transformer-svg-react"] }
}
```

Targets in package.json

```
"source": "src/index.html",
"targets": {
  "default": { "distDir": "./public", "publicUrl": "/app" }
}
```

Quote globs so your shell does not expand them. Entries may be files, directories with a package.json source field, or globs.

RECIPES

Scripts in package.json

```
"scripts": {
  "start": "parcel src/index.html",
  "build": "parcel build src/index.html"
}
```

Static site for a sub-path

```
rm -rf dist .parcel-cache
npx parcel build src/index.html \
  --public-url /docs/ --dist-dir site
```

Library build with watch

```
# package.json: "source", "main", "module"
npx parcel watch
npx parcel build --no-optimize
```

Test on a phone over HTTPS

```
npx parcel src/index.html --https --host 0.0.0.0
```