<https://chatgpt.com/share/689b284e-72f0-8010-87a7-0565d6ac6c58>

Can I host a typescript webpage on GitHub pages for free?

**ChatGPT said:**

Yes — you can host a TypeScript-based webpage on GitHub Pages for free, but GitHub Pages only serves static files (HTML, CSS, JavaScript), so your TypeScript code must be **compiled to JavaScript** before hosting.

**How to Host a TypeScript Webpage on GitHub Pages**

**1. Create Your Project**

* Make a folder for your project with:
  + index.html
  + style.css (optional)
  + .ts files (your TypeScript source code)

**2. Add TypeScript Compiler**

* Install TypeScript locally:

bash

CopyEdit

npm install typescript --save-dev

* Initialize TypeScript config:

bash

CopyEdit

npx tsc --init

* This creates a tsconfig.json file. Set:

json

CopyEdit

{

"compilerOptions": {

"target": "es6",

"outDir": "./dist",

"rootDir": "./src"

}

}

**3. Compile to JavaScript**

* Place your .ts files in /src.
* Compile:

bash

CopyEdit

npx tsc

* This outputs .js files into /dist.

**4. Link JavaScript in HTML**

* Your index.html should reference the compiled JavaScript, not the .ts file:

html

CopyEdit

<script src="dist/main.js"></script>

**5. Push to GitHub**

* Create a new repository on GitHub.
* Commit your project folder (including dist files).

**6. Enable GitHub Pages**

* Go to **Settings → Pages**.
* Under “Branch,” select:
  + main branch
  + / (root) folder **or** /docs folder (if you put site files there)
* Save settings.

**7. Visit Your Site**

* GitHub will give you a free URL like:

arduino

CopyEdit

https://your-username.github.io/your-repo-name/

✅ **Why this works:** GitHub Pages just serves static files. Once TypeScript is compiled to JavaScript, it behaves like any normal webpage.  
💡 You can automate the build with GitHub Actions so every push compiles your .ts files automatically.

If you want, I can give you a **ready-to-use GitHub Pages + TypeScript template** so you don’t have to set up the build manually. Would you like that?

**You said:**

Can I use the same or similar method used for react in this tutorial? https://github.com/gitname/react-gh-pages/blob/master/README.md

**ChatGPT said:**

Yes — you can use a method similar to the React deployment tutorial on GitHub Pages to host a **TypeScript webpage**, because the core process is the same: build your project into static files and deploy them for free.

**How it works for React (from the tutorial)**

The react-gh-pages setup typically:

* Uses the **gh-pages npm package** to deploy built files to a gh-pages branch [GitHub](https://github.com/gitname/react-gh-pages?utm_source=chatgpt.com)[LogRocket Blog](https://blog.logrocket.com/gh-pages-react-apps/?utm_source=chatgpt.com).
* Adds a homepage field in package.json pointing to your GitHub Pages URL [GitHub](https://github.com/gitname/react-gh-pages?utm_source=chatgpt.com)[LogRocket Blog](https://blog.logrocket.com/gh-pages-react-apps/?utm_source=chatgpt.com).
* Defines scripts like:

json

CopyEdit

"predeploy": "npm run build",

"deploy": "gh-pages -d build"

* + npm run build creates static files.
  + npm run deploy pushes the built files to the gh-pages branch [GitHub](https://github.com/gitname/react-gh-pages?utm_source=chatgpt.com)[LogRocket Blog](https://blog.logrocket.com/gh-pages-react-apps/?utm_source=chatgpt.com).

**Adapting that for a TypeScript-only webpage**

If you're building your site directly in TypeScript (without React), you can follow the same idea:

| **Step** | **What to do** |
| --- | --- |
| **1. Compile to JavaScript** | Use tsc (TypeScript compiler) to convert .ts files into .js. Configure your tsconfig.json with outDir, e.g., "dist". |
| **2. Add to package.json** | Include something like: |
|  | ```json |
| { |  |
| "homepage": "https://yourusername.github.io/your-repo", |  |
| "scripts": { |  |

json

CopyEdit

"build": "tsc",

"predeploy": "npm run build",

"deploy": "gh-pages -d dist"

},  
"devDependencies": {  
"gh-pages": "^x.y.z",  
"typescript": "^x.y.z"  
}  
}

|

CopyEdit

| \*\*3. Install packages\*\* | Run `npm install --save-dev typescript gh-pages`. |

| \*\*4. Compile & Deploy\*\* | Run `npm run deploy`. This:

- Runs `tsc` to generate JS files in `dist`,

- Pushes `dist` to the `gh-pages` branch, making the site live. |

| \*\*5. Configure GitHub Pages\*\* | In your repo's \*\*Settings → Pages\*\*, choose to serve from `gh-pages` branch root (exactly like the tutorial) :contentReference[oaicite:3]{index=3}. |

---

### Why it works & benefits

- \*\*Why it works\*\*: GitHub Pages only hosts \*\*static files\*\* (HTML, JS, CSS). Whether the files come from TypeScript or React, as long as they're compiled to static content, they're compatible.

- \*\*Benefits\*\*:

- No extra server or cost — it's free.

- Clear workflow — build then deploy.

- Easy automation with `predeploy`/`deploy` scripts.

---

**ChatGPT said:**

Your package.json is currently set up for an **Expo (React Native)** project — that’s different from a plain TypeScript webpage.  
You *can* still host it on GitHub Pages, but Expo exports web builds into a dist folder (or whatever you choose), and you’ll need to add the gh-pages deployment scripts.

Here’s a modified version of your package.json so it works with the **react-gh-pages** style deployment:

{

"name": "bolt-expo-starter",

"main": "expo-router/entry",

"version": "1.0.0",

"private": true,

"homepage": "https://YOUR\_GITHUB\_USERNAME.github.io/YOUR\_REPO\_NAME",

"scripts": {

"dev": "EXPO\_NO\_TELEMETRY=1 expo start",

"build:web": "expo export --platform web --output-dir dist",

"predeploy": "npm run build:web",

"deploy": "gh-pages -d dist",

"lint": "expo lint"

},

"dependencies": {

"@expo/vector-icons": "^14.1.0",

"@lucide/lab": "^0.1.2",

"@react-navigation/bottom-tabs": "^7.4.6",

"@react-navigation/native": "^7.0.14",

"expo": "^53.0.0",

"expo-blur": "~14.1.3",

"expo-camera": "~16.1.5",

"expo-constants": "~17.1.3",

"expo-font": "~13.2.2",

"expo-haptics": "^14.1.4",

"expo-linear-gradient": "^14.1.5",

"expo-linking": "~7.1.3",

"expo-location": "^18.1.6",

"expo-router": "~5.0.2",

"expo-splash-screen": "~0.30.6",

"expo-status-bar": "~2.2.2",

"expo-symbols": "~0.4.3",

"expo-system-ui": "~5.0.5",

"expo-web-browser": "~14.1.5",

"lucide-react-native": "^0.475.0",

"react": "19.0.0",

"react-dom": "19.0.0",

"react-native": "0.79.1",

"react-native-gesture-handler": "~2.24.0",

"react-native-maps": "^1.25.3",

"react-native-reanimated": "~3.17.4",

"react-native-safe-area-context": "5.3.0",

"react-native-screens": "~4.10.0",

"react-native-svg": "15.11.2",

"react-native-url-polyfill": "^2.0.0",

"react-native-web": "^0.20.0",

"react-native-webview": "13.13.5"

},

"devDependencies": {

"@babel/core": "^7.25.2",

"@types/react": "~19.0.10",

"typescript": "~5.8.3",

"gh-pages": "^6.1.1"

}

}

**What Changed**

1. **homepage field**
   * Replace YOUR\_GITHUB\_USERNAME and YOUR\_REPO\_NAME with your actual GitHub username and repository name.
   * This ensures relative paths work when the site is hosted on GitHub Pages.
2. **gh-pages added**
   * Added "gh-pages": "^6.1.1" to devDependencies.
3. **Deployment scripts**
   * predeploy: Runs the Expo web build before deploying.
   * deploy: Pushes the dist folder to the gh-pages branch.
4. **Output folder set**
   * Changed expo export --platform web to expo export --platform web --output-dir dist so the deploy script knows where to find the files.

**Deployment Steps**

1. Install new dependency:

bash

CopyEdit

npm install --save-dev gh-pages

1. Build and deploy:

bash

CopyEdit

npm run deploy

1. In GitHub repo settings:
   * Go to **Settings → Pages**.
   * Select the gh-pages branch and root folder.
   * Save.
2. Visit your site at:

cpp

CopyEdit

https://YOUR\_GITHUB\_USERNAME.github.io/YOUR\_REPO\_NAME

If you want, I can also give you a **version that deploys automatically with GitHub Actions** so you never run npm run deploy manually. That would make this totally hands-off once you push code.

**You said:**

Yes please

**ChatGPT said:**

Here’s how to make your **Expo + TypeScript** app automatically deploy to GitHub Pages using **GitHub Actions** — no more running npm run deploy manually.

**1. Keep your updated package.json**

Use the one I gave you earlier with:

* "homepage" set
* "predeploy" and "deploy" scripts
* "gh-pages" in devDependencies

**2. Create the GitHub Actions workflow**

Inside your project folder, make this file:  
**.github/workflows/deploy.yml**

name: Deploy to GitHub Pages

on:

push:

branches:

- main # Change to your default branch if different

permissions:

contents: write

jobs:

build-and-deploy:

runs-on: ubuntu-latest

steps:

- name: Checkout repository

uses: actions/checkout@v4

- name: Setup Node.js

uses: actions/setup-node@v4

with:

node-version: 20

- name: Install dependencies

run: npm install

- name: Build web version

run: npm run build:web

- name: Deploy to GitHub Pages

uses: peaceiris/actions-gh-pages@v4

with:

github\_token: ${{ secrets.GITHUB\_TOKEN }}

publish\_dir: ./dist

**3. How it works**

* **Trigger**: Runs every time you push to main (you can change that branch name).
* **Node setup**: Uses Node 20 to match modern Expo builds.
* **Build step**: Runs npm run build:web to export your Expo app to the dist folder.
* **Deploy step**: Publishes dist to the gh-pages branch automatically.

**4. Enable GitHub Pages**

* Go to your repo’s **Settings → Pages**
* Under “Source,” pick gh-pages branch, / (root) folder.
* Save.

**5. Deploy flow**

1. Push code to main
2. GitHub Actions builds and deploys automatically
3. Site updates in ~1–2 minutes

Do you want me to tweak this so that it **only deploys when you change web-related code** and not when you push mobile-only changes?  
That way it won’t run unnecessarily every time you commit.