LEARN

How to host a SMART on FHIR app

Hosting a demo is a solved tutorial. Hosting an app a hospital will actually deploy is a compliance problem: your launch URL and redirect URIs sit directly in the PHI path, and the hospital's security team reviews them as such.

Dev hosting is a solved tutorial

A static SMART app on GitHub Pages, Netlify, or Vercel, pointed at the public SMART Launcher or an EHR sandbox, is a perfectly good way to learn the launch handshake. Nothing on this page argues against that. It stops being the answer the moment real credentials and real patients enter the picture.

What "hosting" means for SMART App Launch

In SMART App Launch, the EHR redirects a clinician to your *launch URL*, then back to a registered *redirect URI* with an authorization code. Your host therefore terminates TLS on a request that carries launch context, exchanges and stores tokens, and serves the UI that renders PHI. That host is inside the PHI path — not adjacent to it. Every hospital reviewer treats it that way.

Why GitHub Pages and the SMART Launcher fail hospital review

- No BAA is available, so PHI cannot lawfully flow through them.
- No PHI-grade access control, audit logging, or log retention you can evidence.
- No environment separation, change control, or incident response you can attest to.
- The SMART Launcher is a public test harness with synthetic patients — it is a demonstration tool, not a deployment target.

HIPAA production runtime

A production runtime for a hospital deployment needs, at minimum: a BAA in place, TLS 1.2+ in transit and encryption at rest, least-privilege access control, immutable audit logs of PHI access, documented incident response, U.S. data residency, and a SOC 2 Type II report to hand the reviewer. SoFaaS™ is that runtime: it owns the launch URL, the OAuth exchange, the FHIR traffic, the hosting posture, and the evidence packet, so those controls are inherited rather than rebuilt per vendor.

> **SEE HOW SOFAAS™ HANDLES THIS**
>
> The practical test: can you answer "who has access to the server that saw the token, and show me the log?" If the answer is a public static host, the review ends there. See /security for what hospitals actually ask.

Backend vs static front end

A pure static front end can only do a public SMART client with PKCE — no client secret, no server-side token storage, no server-to-server refresh. Most clinician-facing apps end up needing a confidential client and a backend for token handling, write-back, and audit. Deciding this early avoids re-architecting during a security review. The mechanics are in the auth docs, and the mental model in getting started.

Compose, don't replace

Redox and similar integration engines translate and route data between systems; they are not your app's host. Epic Showroom lists your app; it is not your app's host either. A runtime composes with both — see Showroom vs runtime vs integration platform, or talk to us about your specific stack.

Frequently asked questions

Can I host a SMART on FHIR app on GitHub Pages?

For learning and demos against the public SMART Launcher, yes. For a hospital deployment, no. GitHub Pages cannot be covered by a BAA, gives you no audit logging or access controls over the PHI path, and will not survive a hospital security review.

Is the SMART Launcher enough to demo for a hospital?

It is enough to show the launch mechanics with synthetic patients. It is not enough for a hospital evaluation that involves real credentials, real PHI, or a security questionnaire — those need a production runtime with HIPAA hosting behind it.

Does Epic host my app if I'm on Showroom?

No. Epic Showroom lists the app. Hosting, uptime, TLS, audit logs, the BAA, and the SOC 2 evidence remain the vendor's responsibility — or the responsibility of the runtime the vendor deploys on.

What makes hosting HIPAA-eligible for a SMART app?

A signed BAA with the hosting provider and with you, encryption in transit (TLS 1.2+) and at rest, access control and least privilege, immutable audit logging of PHI access, incident response, and a documented control set you can evidence — typically a SOC 2 Type II report. U.S. data residency is expected by most health systems.

Do I still need my own cloud if I use a compliant runtime?

Usually only for the parts of your product that live outside the EHR launch. The launch URL, redirect URIs, token handling, and FHIR traffic run on the runtime; your own backend can stay where it is, as long as it does not become an undocumented PHI path.

Is this the same as an integration engine (Redox)?

No. An integration engine translates and routes clinical data between systems. A runtime is where your clinician-facing app actually runs and launches from inside the chart. They compose; neither replaces the other.

Related

What is SMART on FHIR? →

How to get your app into Epic →

Epic Showroom vs a SMART on FHIR runtime →

Epic vendor security review →

What is SoFaaS™? →

Have a deal stuck on Epic?

SoFaaS™ is the compliant runtime that ships SMART on FHIR apps into Epic in weeks, not quarters.

Talk to us →