Tiny Websites have no Server

Originalartikel

Backup

<html> A big trend in web services right now is the so-called serverless computing, such as Amazon’s Lambda service. The idea is you don’t have a dedicated server waiting for requests for a specific purpose. Instead, you have one server (such as Amazon’s) listening for lots of requests and on demand, you spin up an environment to process that request. Conceptually, it lets you run a bit of Javascript or some other language “in the cloud” with no dedicated server. A new concept — https://itty.bitty.site — takes this one step farther. The site creates self-contained websites where the content is encoded in the URL itself. Probably the best example is to simply go to the site and click on “About itty bitty.” That page is itself encoded in its own URL. If you then click on the App link, you’ll see a calculator, showing that this isn’t just for snippets of text. While this does depend on the itty.bitty.site web host to provide the decoding framework, the decoding is done totally in your browser and the code is open source. What that means is you could host it on your own server, if you wanted to. At first, this seems like a novelty until you start thinking about it. A small computer with an Internet connection could easily formulate these URLs to create web pages. A bigger computer could even host the itty.bitty server. Then there’s the privacy issue. At first, we were thinking that a page like this would be hard to censor since there is no centralized server with the content. But you still need the decoding framework. However, that wouldn’t stop a sophisticated user from “redirecting” to another — maybe private — decoding website and reading the page regardless of anyone’s disapproval of the content. That might be the most compelling case of all. You can encode something in a URL and then anyone with that URL could read your content even if someone shuts down your servers (or the itty bitty servers). The itty bitty server just hands out some generic JavaScript. The website data is stored as a fragment which — interestingly enough — doesn’t get sent to the server. That means the server doesn’t even get a look at what you are trying to decode. It just provides the

decoding framework and your browser does all the rest of the work locally. We’d love to see someone fork the project and add simple encryption, too. Currently, the text is compressed and base 64 encoded, but anyone with the URL can decode what it says. An encryption key would allow you to send URLs in the clear that only some people could decode and would be very hard to suppress. The itty bitty code itself is an app since you can edit most pages with an edit link at the top right corner. If you don’t like editing in place, the site explains how you can use a generic HTML file or use an online HTML editor, if you prefer. There are limitations. You probably can’t host graphics internally — you’d need an external place to point to pictures. You also can make really long URLs — which means some services like Twitter will cut them off. We figure you could use a URL shortener if you needed to. There’s also a way to make a QR code baked right in. We could see this replacing a server on a Raspberry Pi project. While this isn’t technically serverless computing, it did remind us of how to write code for assistants. </html>