

WebCmd.js Manifest

Neruthes

2023-10-09

1 Introduction

The idea of interpreting web APIs as commands was documented in [an article](#) written in April 2023.

We at this project aim to provide a usable demo for this new paradigm, which may be named "WebCmd".

2 Technical Context

This framework `webcmdjs` may be combined with Nodejs standard HTTP APIs, or some backend frameworks which handle routing.

As a user of this library, you should handle authentication with your own measures. The world of authentication is sophisticated and we will not mandate any paradigm for authentication.

Take the example of SSH. Imagine that your customer Alice is submitting a support ticket in an SSH session. This library provides the `bash` on the surface and you have to implement your own `sshd` which feeds `USER`, `UID`, and `GID` as *environment variables* for the `bash` process.

3 API Design

3.1 Basic Example

3.1.1 Creating Application

```
const webcmd = require('webcmdjs').createApp();

webcmd.define('echo', function (data) {
  // data has cmd/args/req/res/env
  data.res.writeHead(200);
  data.res.end(data.args.text);
});
```

3.1.2 Listening Port

```
const http = require('http');

http.createServer(function (req, res) {
  if (req.uri === '/api/webcmd') {
    // Implement your own token authenticator
    const env = myAuth(req.headers.token);
    webcmd.accept(req, res, env);
  }
}).listen(12345);
```

3.1.3 Sending Request

```
curl -X POST \
  --data '{"cmd": "echo", "args": {"text": "Loerm ipsum dolor sit amet"}}' \
  -H 'token: 1145141919810' \
  'http://localhost:12349/api/webcmd'
```