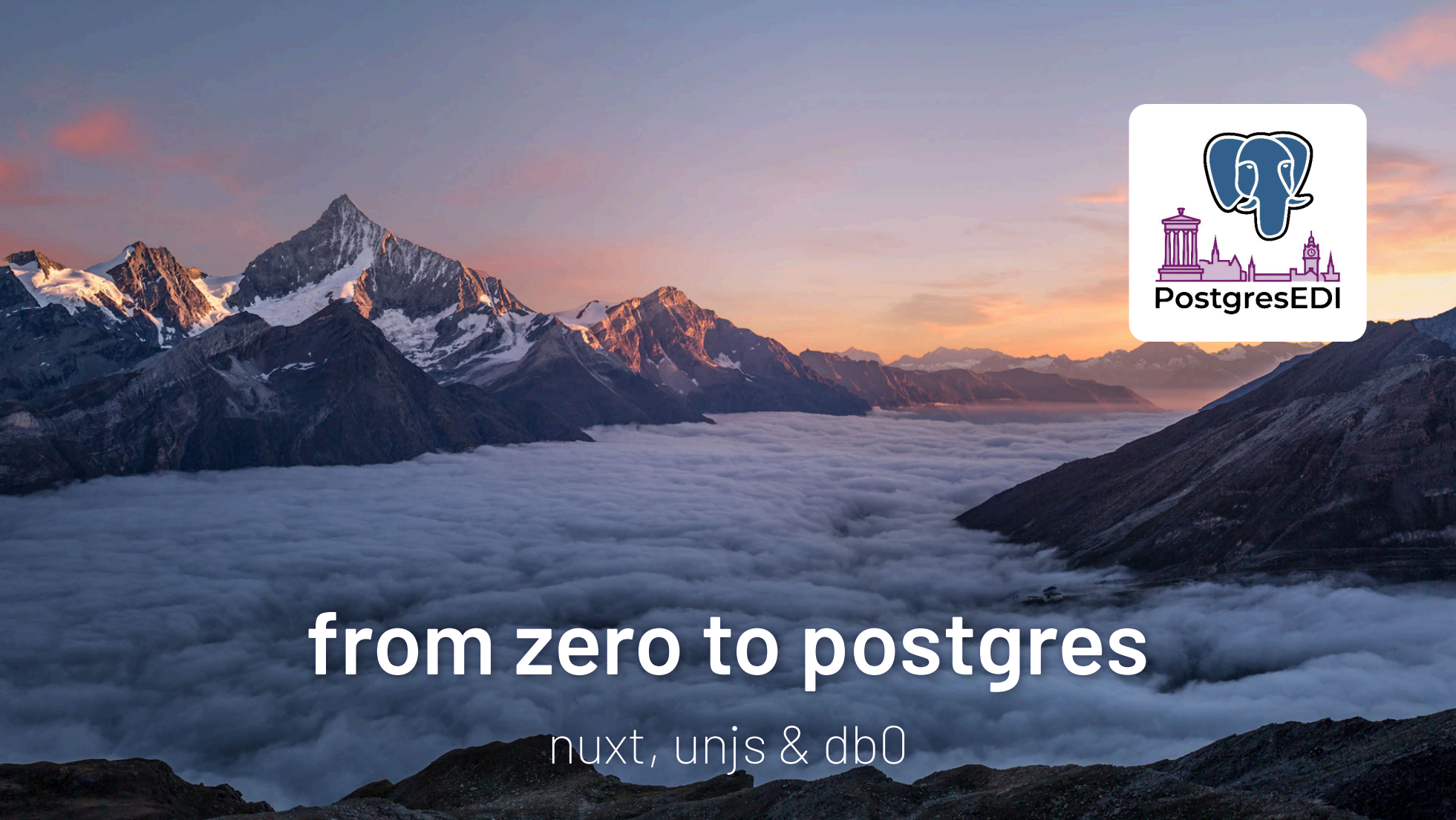


from zero to postgres

nuxt, unjs & db0

Daniel Roe he/him
daniel@roe.dev

@danielroe.dev
@daniel@roe.dev

hello 👋

nuxt.com • npmx.dev • nitro.build • page-speed.dev
regexp.dev • fontaine.sh • pangrum.com • elk.zone • firstcommit.is

Experts

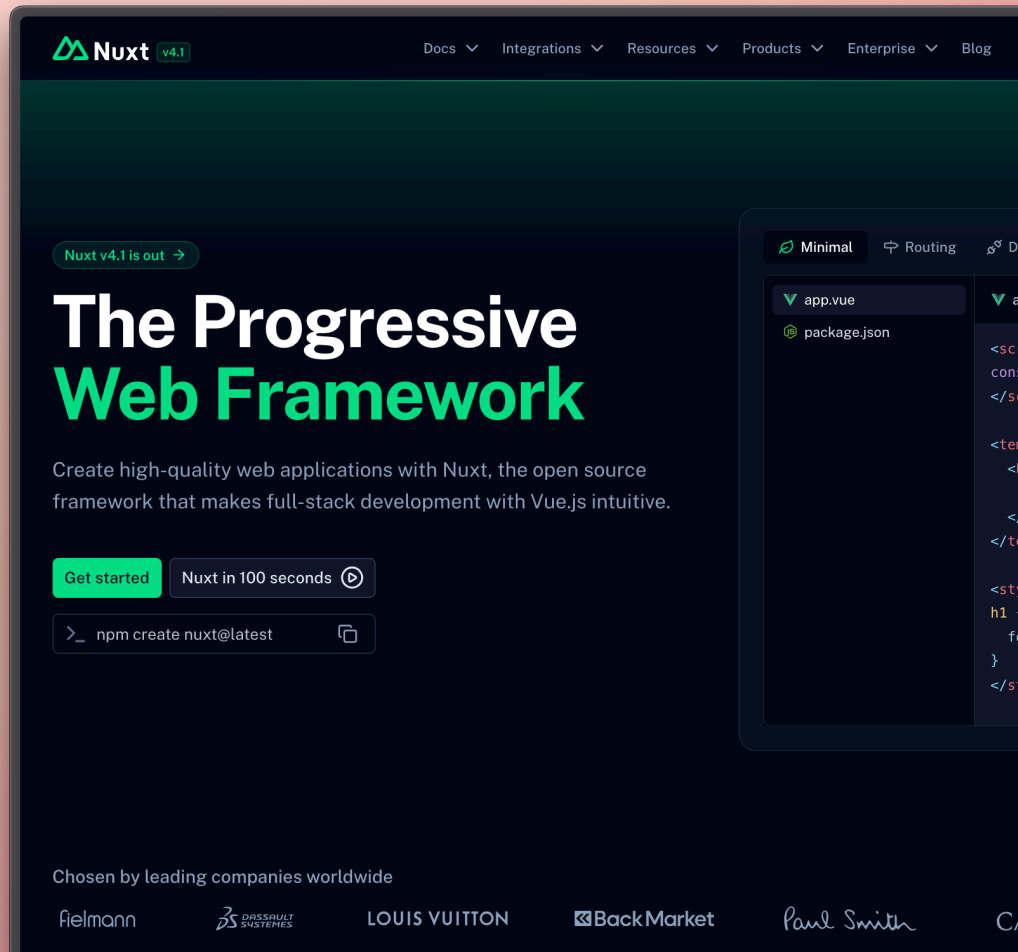
Microsoft Most Valuable Professional

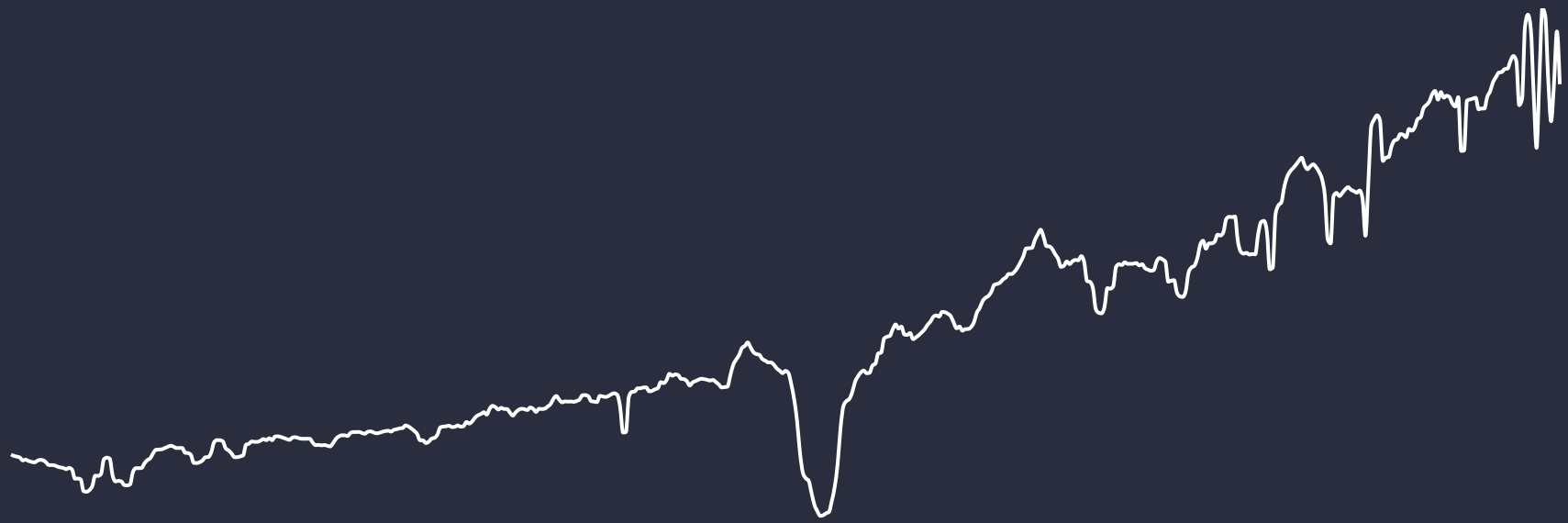
Vercel


roe.dev/live

what is nuxt?

- a **progressive** framework built on **vue.js** and **nitro**
- **zero-effort** start with great dx
- **best practices** built-in
- fully **configurable** & easily **extensible**





60.9 k
STARS

70.8 MILLION
DOWNLOADS

let's build something


introducing unjs
universal · unified · javascript

h3js/h3

Minimal, tree-shakeable HTTP framework

```ts
export default defineEventHandler(async (event) => {
  assertMethod(event, 'POST')

  const query = getQuery(event)
  const body = await readBody(event)

  if (!body) {
    throw createError({ statusCode: 422 })
  }

  return { foo: 'bar' }
})
```

unjs/unstorage

Provider-agnostic, async KV storage API.

```
import { createStorage } from 'unstorage'
import fsDriver from 'unstorage/drivers/fs'
import redisDriver from 'unstorage/drivers/redis'

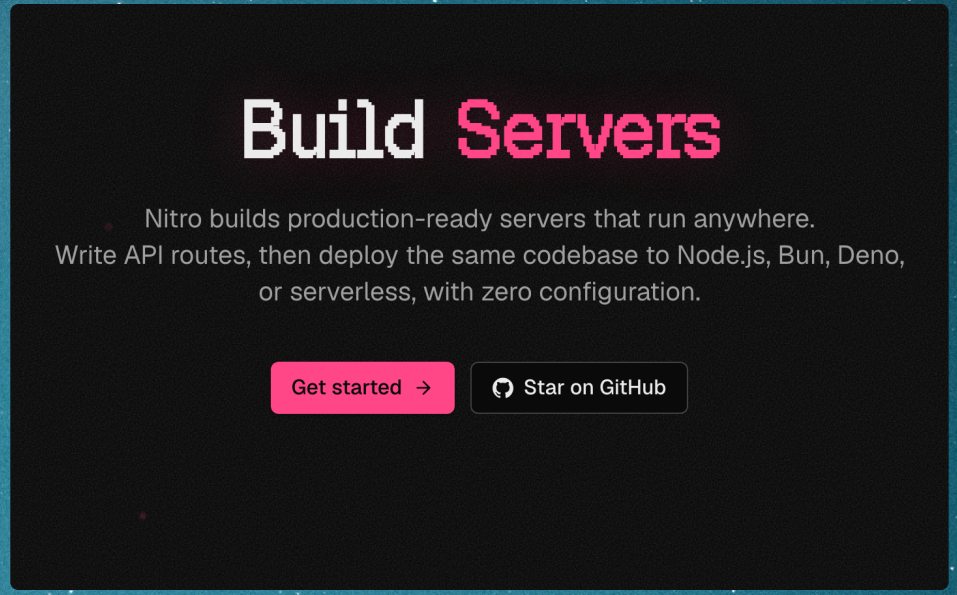
const storage = createStorage()
storage.mount('tmp', fsDriver({ base: '/tmp' }))
storage.mount('cache', redisDriver({
  url: 'redis://daniel:supersecret@roe.dev:6380'
}))

const value = await storage.getItem('cache:my-key')
await storage.setItem('tmp:my-key', { foo: 'bar' })
```

nitrojs/nitro

the **next-generation server toolkit**

- ✦ JavaScript that runs **anywhere**: Node.js, Deno, Bun, workers
- ✦ optimised for **speed** and **low cold start**
- ✦ storage, caching, tasks ... and **databases**



unjs/db0

Lightweight SQL connector

- ✦ one API, many drivers
- ✦ tagged template queries, parameterised for you
- ✦ plays nicely with drizzle, kysely & prisma

unjs/db0

```ts
import { createDatabase } from 'db0'
import neon from 'db0/connectors/neon'

const db = createDatabase(neon({
  url: process.env.DATABASE_URL,
}))

const city = 'edinburgh'
const { rows } = await db.sql`SELECT * FROM meetups WHERE city = ${city}`
```

unjs/db0

- **postgresql**
- **neon**
- **pglite**
- cloudflare hyperdrive (pg)
- planetscale
- mysql2
- cloudflare hyperdrive (mysql)
- cloudflare d1
- libsql / turso
- better-sqlite3
- node:sqlite
- bun:sqlite

db0 in nuxt

```ts
// nuxt.config.ts
export default defineNuxtConfig({
  nitro: {
    experimental: { database: true },
    database: {
      default: {
        connector: 'neon',
        options: { url: process.env.DATABASE_URL },
      },
    },
  },
})
```

db0 in nuxt

```ts
// server/api/meetups.get.ts
export default defineEventHandler(async () => {
  const db = useDatabase()
  const { rows } = await db.sql`SELECT * FROM meetups`
  return rows
})
```

... but I need a database


build with others

picking a north star

- aim for progressive adoption
  - discoverable
  - incrementally adoptable
  - backwards-compatible
- build with others
- don’t become the bottleneck



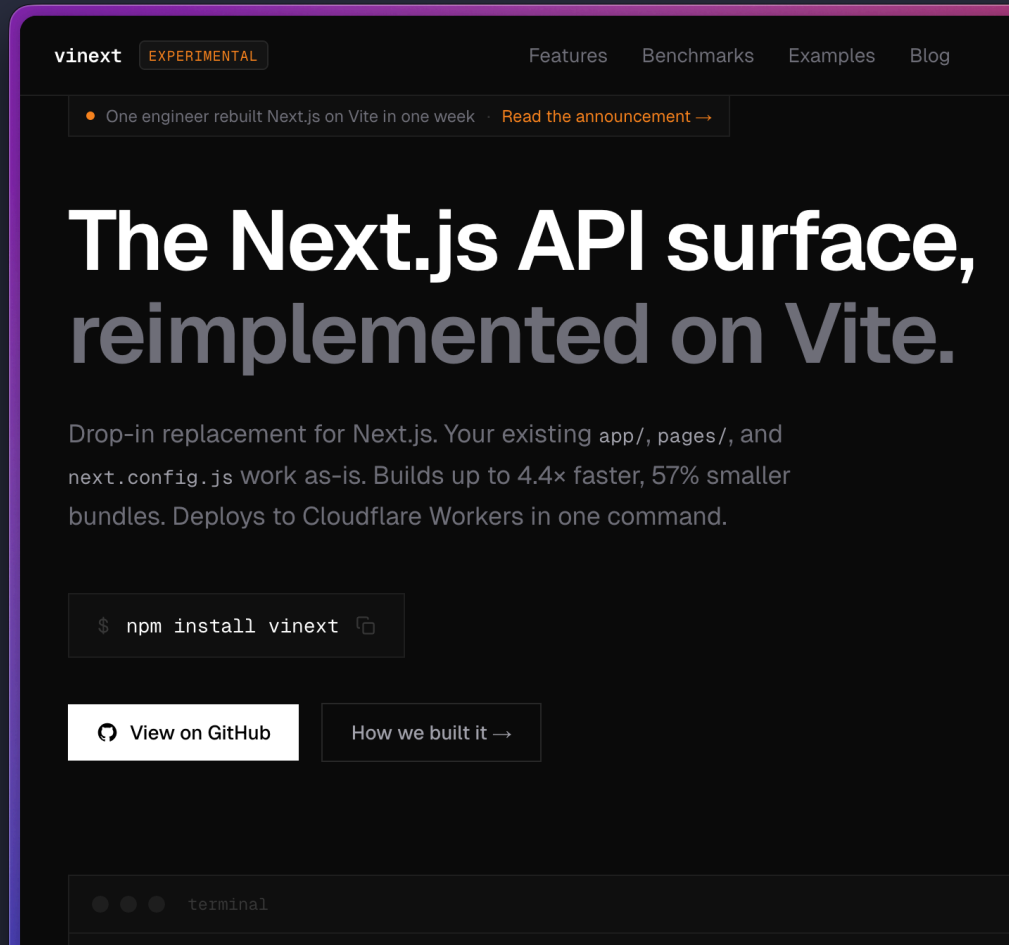
'centre-bounded'



some open source principles

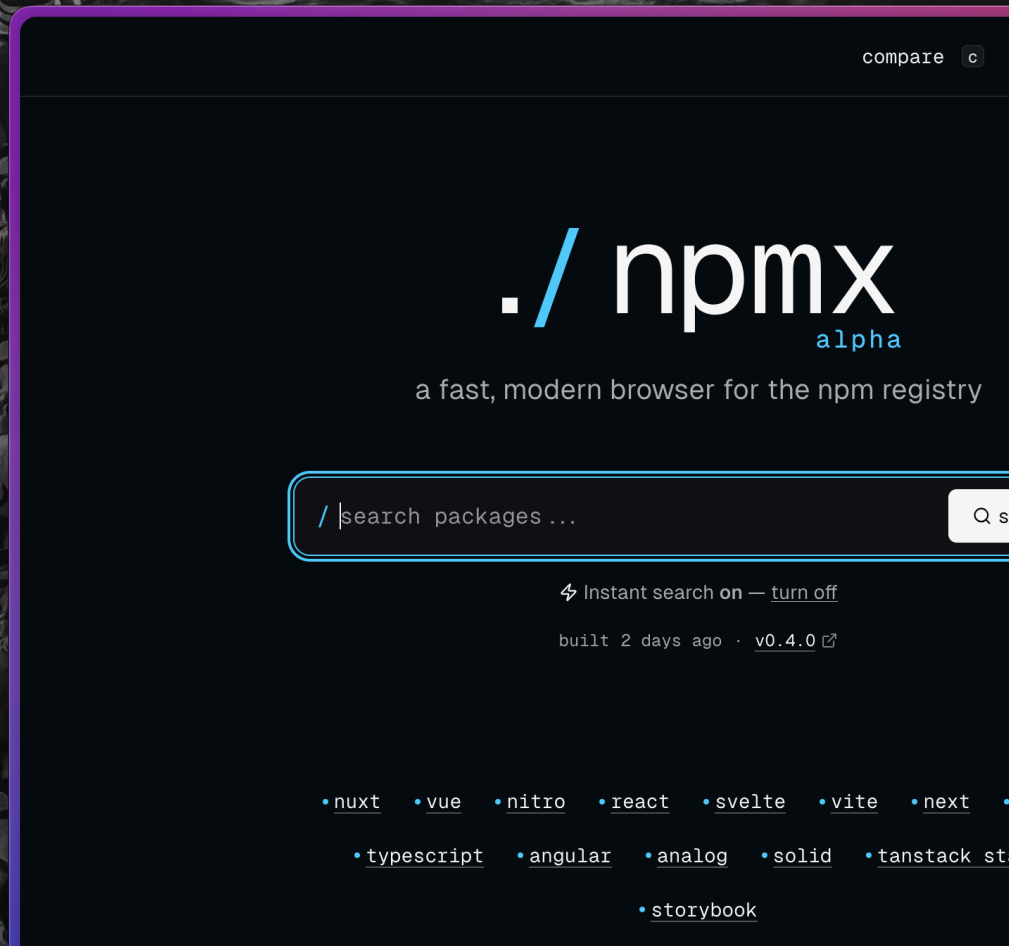
- humanity - **ai-policy.dev**
- inclusion
- empowerment - shoutout to **DefinitelyTyped**

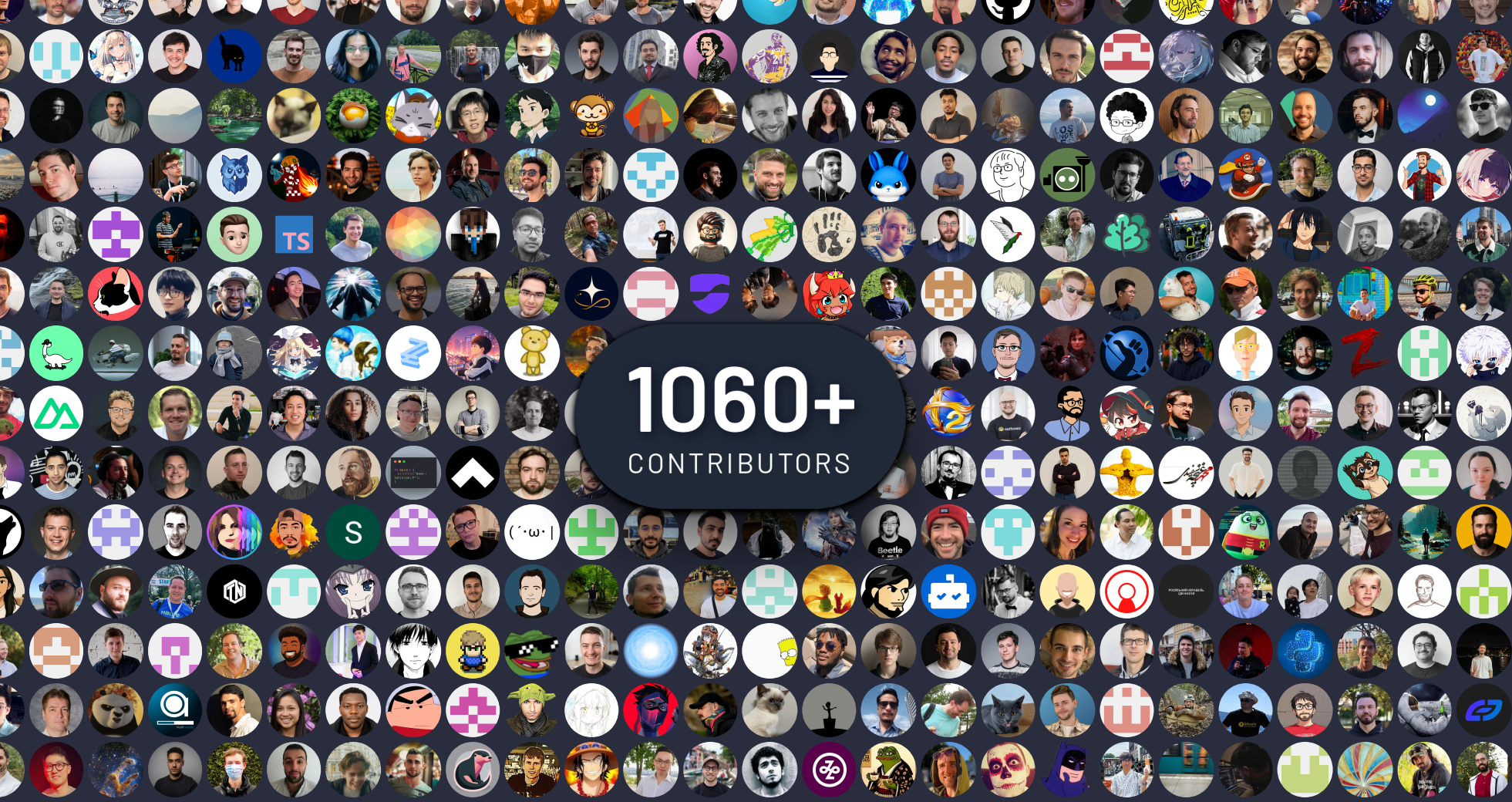
the 10x developer



~~the 10x developer~~

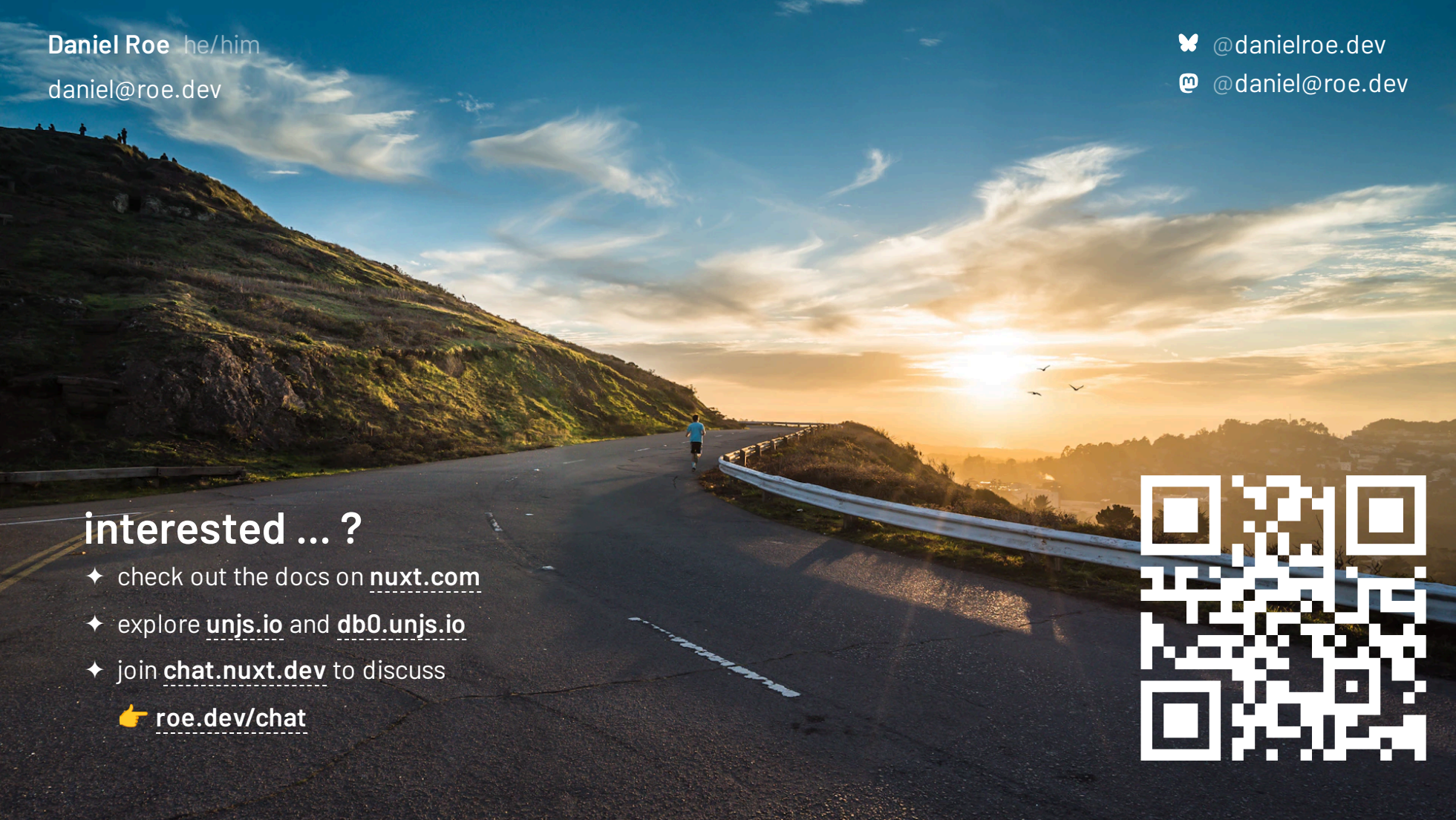
- ✦ we need 10x **teams**
- ✦ people who care about the same problem
- ✦ ... who iterate together
- ✦ ... and make each other better




1060+
CONTRIBUTORS

Daniel Roe he/him
daniel@roe.dev

@danielroe.dev
@daniel@roe.dev



interested ... ?

- ✦ check out the docs on **nuxt.com**
- ✦ explore **unjs.io** and **db0.unjs.io**
- ✦ join **chat.nuxt.dev** to discuss

  👉 **roe.dev/chat**