
TS Congress
Making Magic: Building a TypeScript-First Framework

hello 👋

🦄 regexp.dev • 🅰️ fontaine.sh • 🦌 @elk • @server • DCR roe.dev

Google GDE • MVP Microsoft Most Valuable Professional

Daniel Roe
daniel@roe.dev

@danielcroe
@daniel@roe.dev

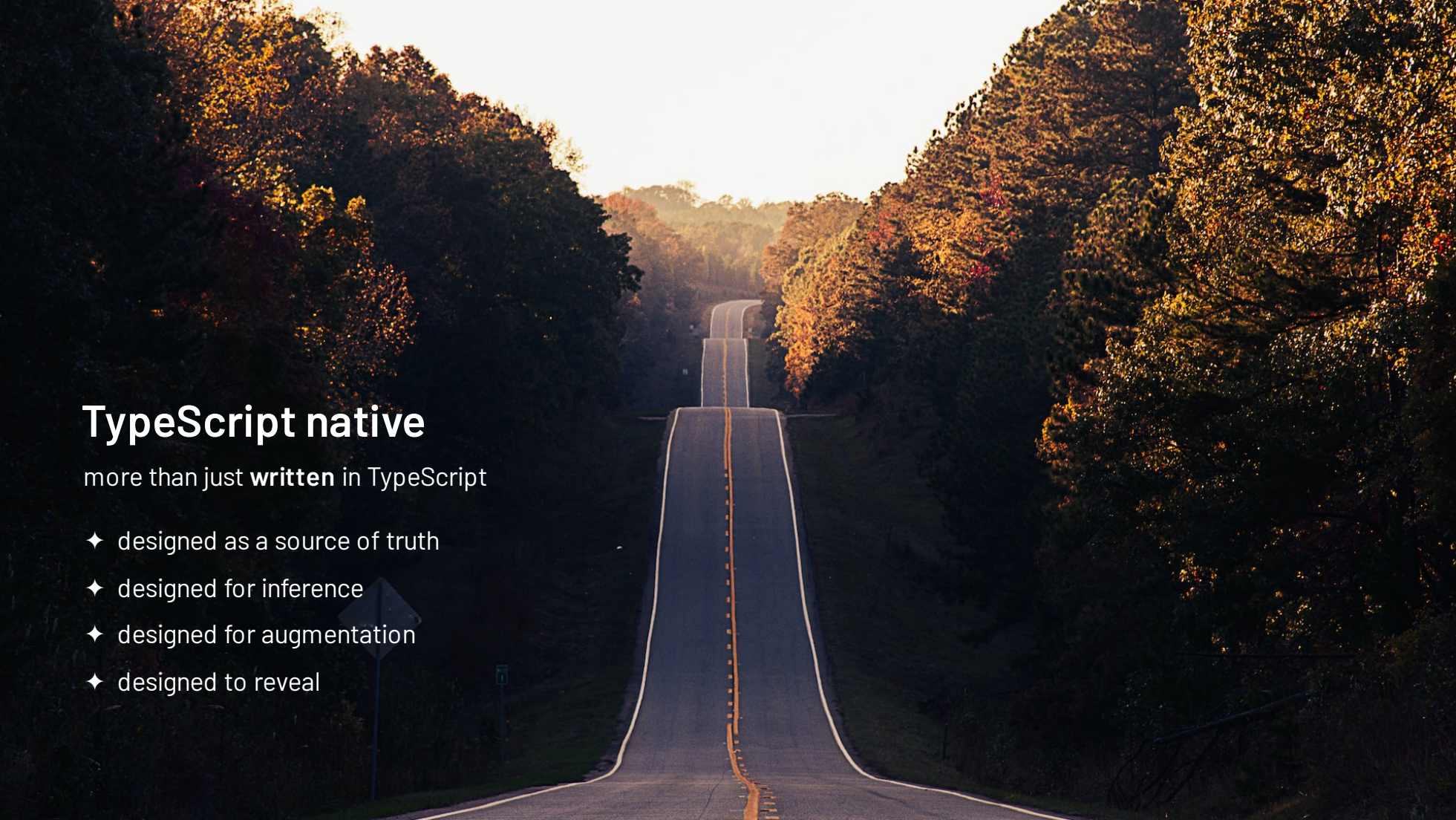

YETI

What is Nuxt?

- a **progressive** framework built on **Vue.js**
- **zero-effort** start with great DX
- **best practices** built-in
- fully **configurable** & easily **extensible**

Magic ✨

- ✦ context (≈ friction)
- ✦ minimalism (≈ distraction)



TypeScript native

more than just **written** in TypeScript

- ✦ designed as a source of truth
- ✦ designed for inference
- ✦ designed for augmentation
- ✦ designed to reveal

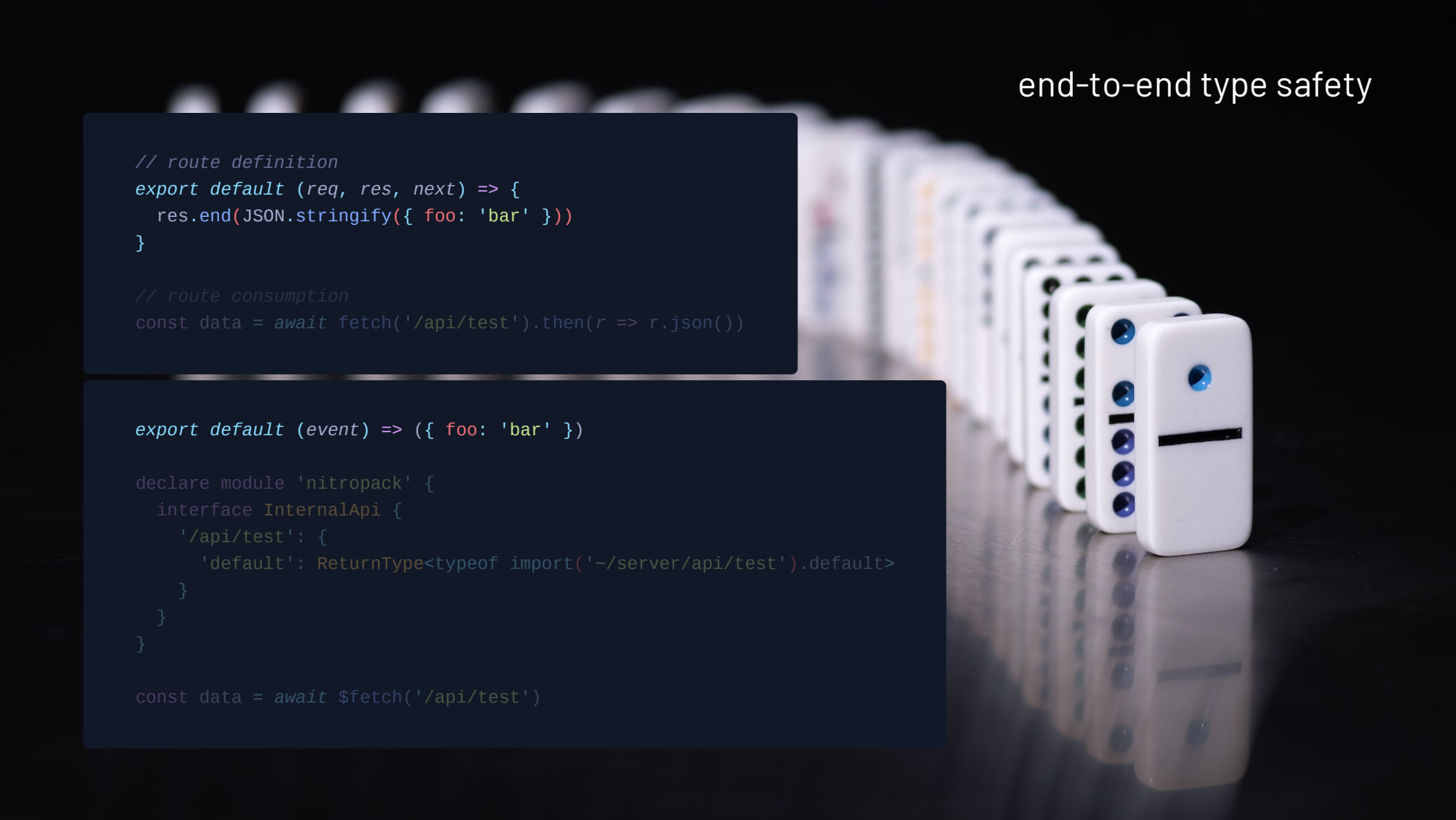
Designed as a source of truth

- reading vs writing `tsconfig.json`
- editor plugins vs native TypeScript

Designed for inference

- ✦ end-to-end type safety
- ✦ automatic context injections

end-to-end type safety



```ts
// route definition
export default (req, res, next) => {
  res.end(JSON.stringify({ foo: 'bar' }))
}

// route consumption
const data = await fetch('/api/test').then(r => r.json())
```

```ts
export default (event) => ({ foo: 'bar' })

declare module 'nitropack' {
  interface InternalApi {
    '/api/test': {
      'default': ReturnType<typeof import('~/server/api/test').default>
    }
  }
}

const data = await $fetch('/api/test')
```

context injections

```ts
export default (ctx, inject) => inject('auth', { login: () => { /* */ } })

export default {
  asyncData({ $auth }) { /* do something with $auth */ }
}
```

```ts
export default defineNuxtPlugin(nuxtApp => ({
  provide: {
    auth: { login: () => { /* */ } }
  }
}))

type InjectionType<A extends Plugin> = A extends Plugin<infer T> ? Decorate<T> : unknown
type NuxtAppInjections = InjectionType<typeof import("../../plugins/some-plugin").default>

declare module '#app' {
  interface NuxtApp extends NuxtAppInjections { }
}

useNuxtApp().$auth.login()
```

Designed for augmentation

modules ecosystem and **extensible** philosophy

- ✦ interface augmentation
- ✦ type utilities for extension

```ts
export default defineNuxtModule({
  setup() {
    addTypeTemplate({
      filename: 'test.d.ts',
      async getContents() {
        return `
          declare module '#app' {
            interface PageMeta {
              customMeta: string
            }
          }
        `
      }
    })
  },
})
```

```ts
export default defineNuxtModule({
  setup(options, nuxt) {
    nuxt.hook('prepare:types', context => {
      const { tsConfig, references, declarations } = context
      // direct access to generated `tsConfig` and
      // generated `nuxt.d.ts` references/declarations
    })
  },
})
```

Exposing the truth

- app config
- runtime config and environment variables
- server and client context
- middleware, layouts
- auto-imports
- aliases
- components

Interested ... ?

- ✦ check out the docs on **nuxt.com**
- ✦ follow **@nuxt_js** on X or **@nuxt@webtoo.ls** on Mastodon
- ✦ join **chat.nuxt.dev** to discuss

Daniel Roe
daniel@roe.dev

@danielcroe
@daniel@roe.dev