Generate pages on demand
Incremental Static Generation (ISG) With Nuxt.js and Layer0

Layer0 by Limelight NETWORKS

Rishi Raj Jain

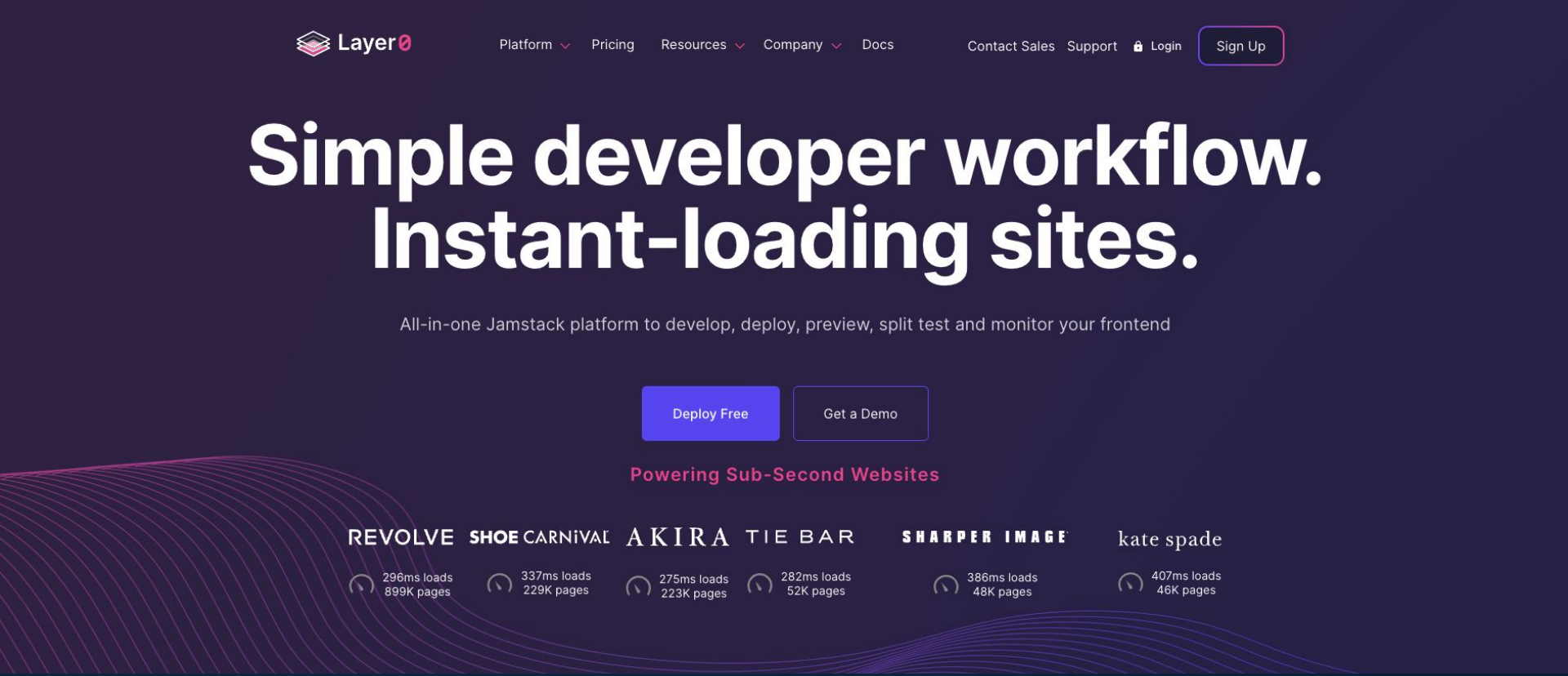
Solutions Engineer at Layer0 by Limelight

Storyblok **Ambassador**

What to expect from this talk.

- What is ISG?
- Benefits of ISG
- Drawbacks of ISG
- Implementing ISG with Nuxt.js and Layer0

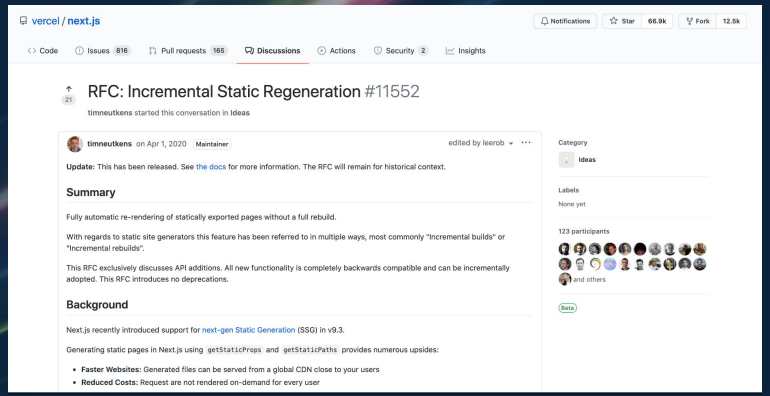
What is Layer0?



What is Layer0? All-in-one Jamstack platform.

Layer0 by Limelight Networks

01 – What is ISG?



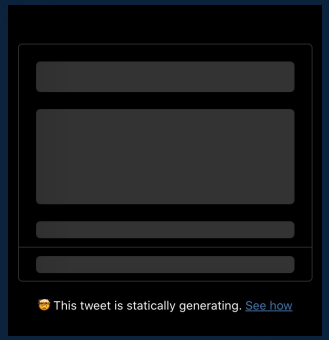
- Generate static pages on demand,
- “Best of both worlds”
  - Builds only the pages you need,
  - Fast “static” page loads after the first visit
- Issues
  - Feature of your infrastructure and your framework,
  - Is limited to Next.js on certain platforms

01 – What is ISG?

Incremental Static Generation

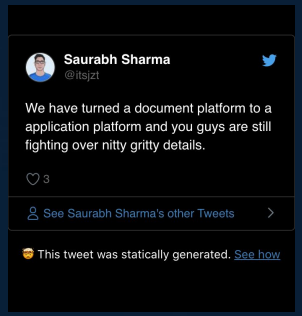
- **ISG on first visit when set to fallback: true**

Browser requests a new page that has not yet been visited



CDN quickly returns a universal “fallback” page with placeholder data...

...while fallback is being displayed, page’s static build process is run

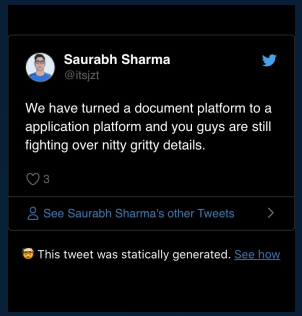


Build completes and fallback loads the static JSON data displaying the final page.

Source: **https://static-tweet.vercel.app/**

Incremental Static Generation

- ISG on subsequent visits when set to fallback: true

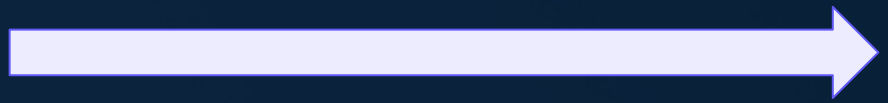


Future visits will get the statically built page HTML (no waiting).

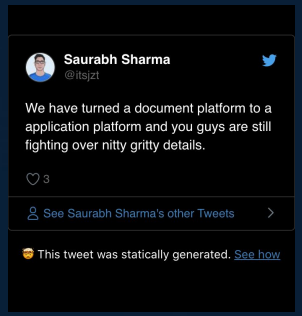
Incremental Static Generation

- **ISG on first visit when set to fallback: blocking**

Browser requests a new page that has not yet been visited

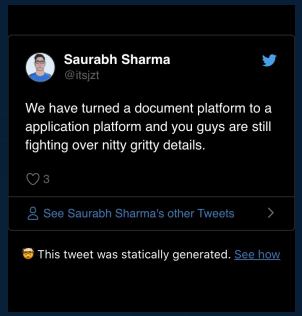


Browser waits for the SSR generated response



Incremental Static Generation

- ISG on subsequent visits when set to fallback: blocking



Future visits will get the statically built page HTML (no waiting).

02 - Benefits of ISG

- Pages on-demand,
- Minimal static build,
- No more re-building the website,
- Faster deployments,
- Easy cache purging with Layer0 REST API

02 - Benefits of ISG

03 - Drawbacks of ISG

- Fallback first for the first user,
- Some users might run into stale data,
- Breaks immutability
    - But not with Layer0 as you can choose whether to keep the cache between deployments

03 - Drawbacks of ISG

04 - Implementing ISG with Nuxt.js and Layer0

04 - In Action: ISG with Nuxt.js and Layer0

- Setup API Routes with Nuxt,
- Setup dynamic routes with Nuxt,
    - Cache dynamic routes on the edge
- Configure Layer0 EdgeJS

04 - Steps to implement ISG with Nuxt.js and Layer0

api/blogs.js

```
const { parse } = require('rss-to-json')
const { Router } = require('express')
const router = Router()

router.use('/blogs/:username.json', async (req, res) => {
  const slug = req.params.username
  let errorResponse = (res) => {
    res.writeHead(404, { 'Content-Type': 'application/json' })
    res.end(
      JSON.stringify({
        code: 0,
      })
    )
  }
  try {
    let rss = await parse(`https://medium.com/feed/@${slug}`)
    let resp = JSON.stringify(rss, null, 3)
    if (!resp) {
      return errorResponse(res)
    }
    resp = JSON.parse(resp)
    res.writeHead(200, { 'Content-Type': 'application/json' })
    res.end(
      JSON.stringify({
        resp,
        code: 1,
      })
    )
  } catch {
    return errorResponse(res)
  }
})

module.exports = router
```

api/index.js

```
const express = require('express')
const app = express()

const hello = require('./blogs')
app.use(hello)

if (require.main === module) {
  const port = 3001
  app.listen(port, () => {
    console.log(`API server listening on port ${port}`)
  })
}

module.exports = app
```

04.1 - Setup API Routes with Nuxt

pages / blogs / _slug.vue

```
async asyncData({ params, redirect }) {
  let link = process.env.API_URL
  if (typeof window !== 'undefined') link = window.location.origin
  let blogsData = await fetch(`${link}/api/blogs/${params.slug}.json`).then((res) => res.json())
  if (blogsData['code'] == 0) redirect(404, '/error')
  return {
    resp: blogsData['resp'],
    slug: params.slug,
    link,
  }
},
```

04.2 - Setup dynamic routes with Nuxt

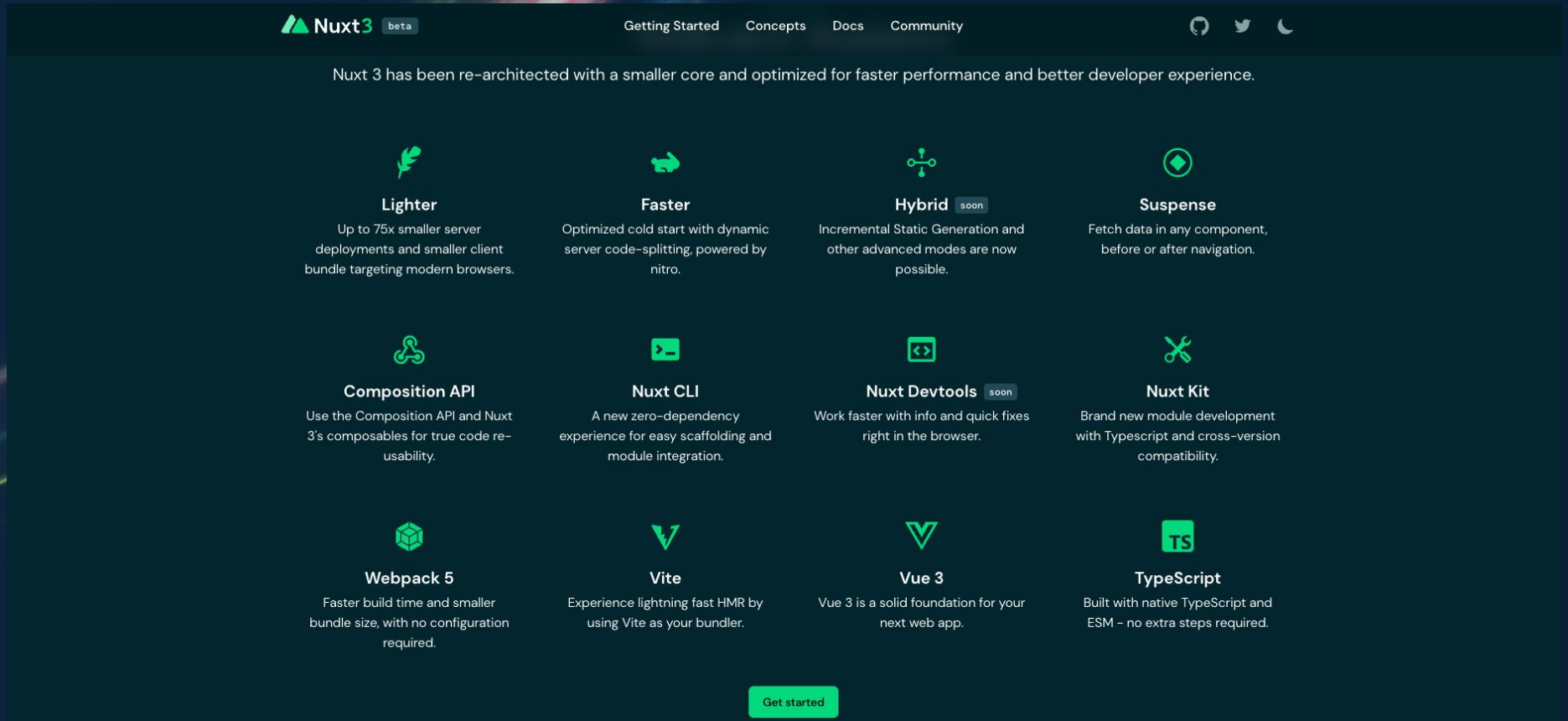
```
mounted() {
  if (typeof window !== 'undefined' && window.__client__ === true) {
    window.__client__ = false
    console.log('Client Side Transition, Populating the cache...')
    fetch(`/blogs/${this.slug}`)
  }
},
```

04.3 - Cache dynamic routes on the edge

```
15    .get('/blogs/:username', ({ serveStatic, cache, renderWithApp }) => {
16      cache({
17        edge: {
18          maxAgeSeconds: 60 * 60 * 24 * 365, // keep the incrementally generated page for a year
19          staleWhileRevalidateSeconds: 1, // revalidate the data on page every second
20        },
21        browser: false,
22      })
23      serveStatic('dist/blogs/:username.html', {
24        // When the user requests a page that is not already statically rendered, fall back to SSR.
25        onNotFound: () => renderWithApp(),
26      })
27    })
28    .get('/api/blogs/:username.json', ({ serveStatic, cache, renderWithApp }) => {
29      cache({
30        edge: {
31          maxAgeSeconds: 60 * 60 * 24, // cache at the edge for 24 hours
32        },
33      })
34      serveStatic('dist/blogs/:username.json', {
35        // When the user requests data that is not already statically rendered, fall back to SSR.
36        onNotFound: () => renderWithApp(),
37      })
38    })
```

04.4 - Configure Layer0 EdgeJS

https://rishi-raj-jain-nuxt-isg-default.layer0.link/



Thank You!