Nuxt Fundamentals

SEO & Metadata

Peter Kassenaar –
info@kassenaar.com

SEO & Metadata

Adding invisible data to the app’s headers to be crawled

SEO & Page meta tags

- Nuxt is great for server sided rendering
  - Render on the server? Use **`npm run generate`** (=default script)
  - Pages are already processed on the server
  - *Faster for the client, better for Search Enging Crawling*
- Multiple options:
  - 1. Update `nuxt.config.ts`
  - 2. Use/override in a specific page, for instance using `useHead()`
  - 3. Use Vue-specific components for <Head>, `<Title>` and so on, to override defaults
    - OR the ones you set in `nuxt.config.ts`
    - Note the Capital in the beginning of the component names. This indicates that they are indeed Vue components and not regular HTML tags.
    - https://nuxt.com/docs/getting-started/seo-meta#components

Static Metadata

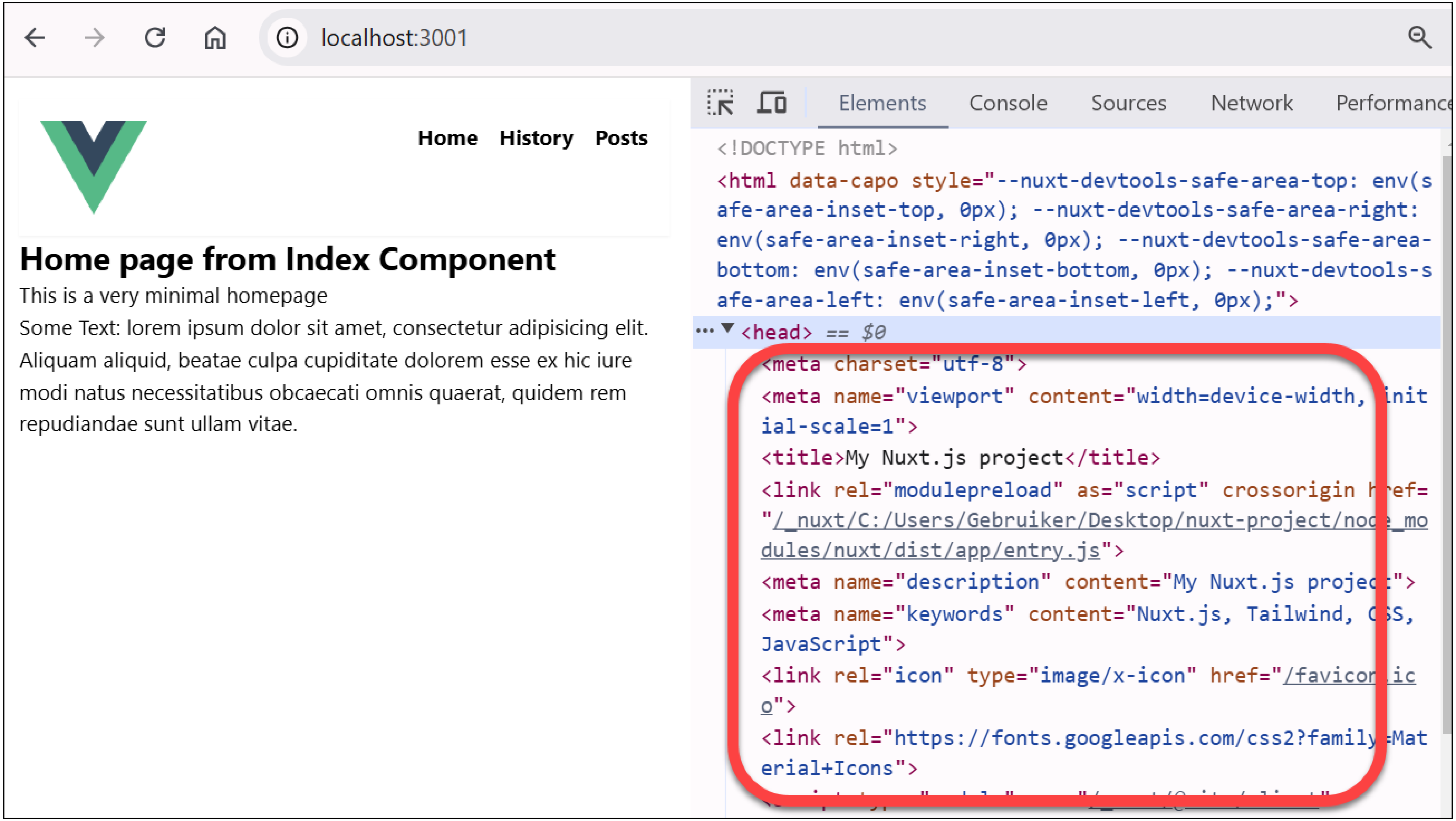
- You can use `nuxt.config.ts` to configure static metadata
  - Things that do not change on page rerender or –refresh
- Very useful for things like
  - `charset`
  - `viewport` width
  - Default bundled `stylesheets`
  - `keyword/description` that doesn’t change for different pages
  - Bundled `favIcon`
  - And more…

1. Update nuxt.config.ts

```
// https://nuxt.com/docs/api/configuration/nuxt-config
export default defineNuxtConfig({
    …
    app: {
        head: {
            title: 'My Nuxt.js project', // static title + meta tags
            meta: [
                {charset: 'utf-8'},
                {name: 'description', content: 'My Nuxt.js project'},
                {name: 'keywords', content: 'Nuxt.js, Tailwind, CSS, JavaScript'},
                {name: 'viewport', content: 'width=device-width, initial-scale=1'}
            ],
            link: [
                {rel: 'icon', type: 'image/x-icon', href: '/favicon.ico'},
                {rel: 'https://fonts.googleapis.com/css2?family=Material+Icons'}
            ]
        }
    }
})
```

https://nuxt.com/docs/getting-started/seo-meta#defaults

Result in browser DevTools



Home History Posts

Home page from Index Component

This is a very minimal homepage

Some Text: lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquam aliquid, beatae culpa cupiditate dolorem esse ex hic iure modi natus necessitatibus obcaecati omnis quaerat, quidem rem repudiandae sunt ullam vitae.

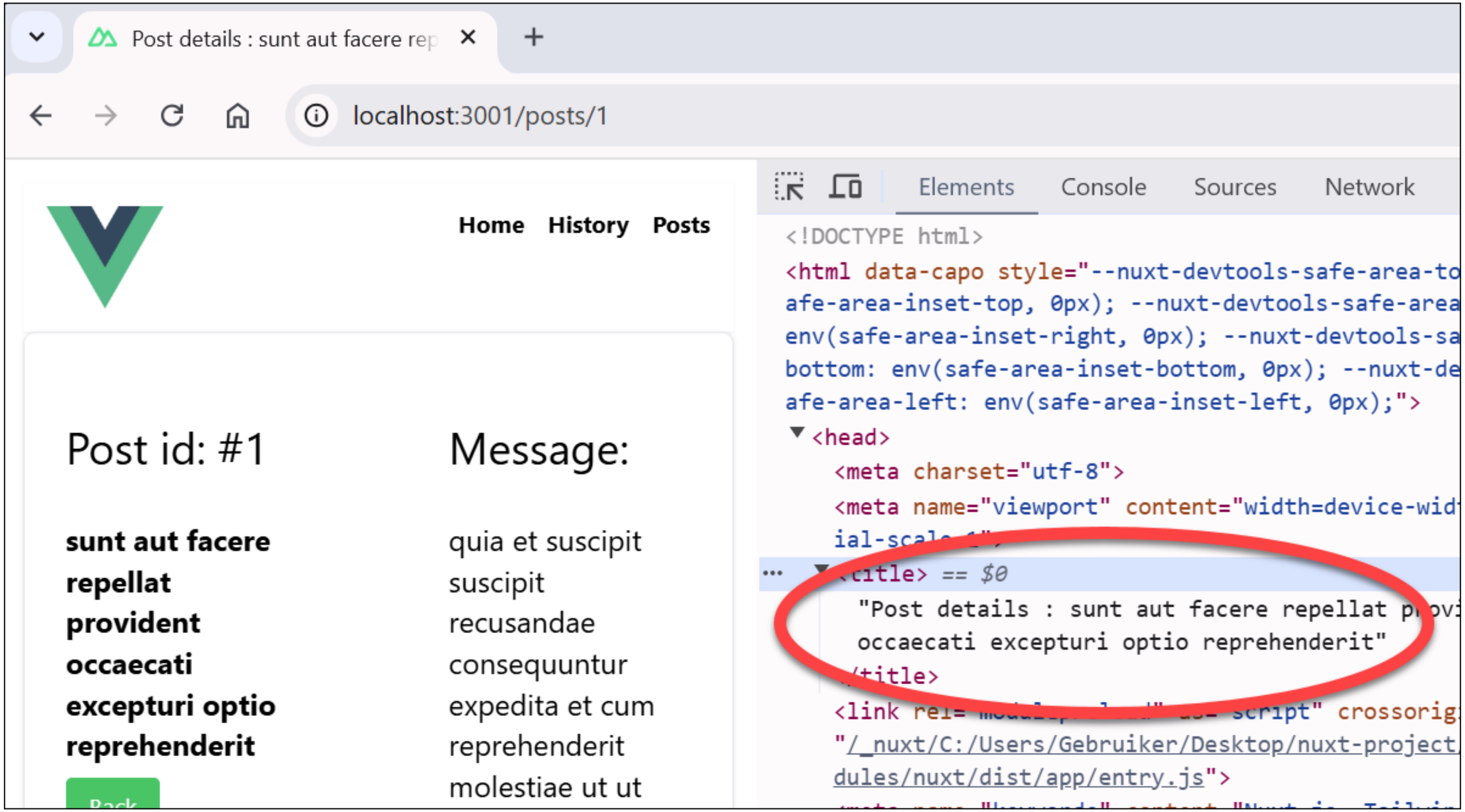
```
<!DOCTYPE html>
<html data-capo style="--nuxt-devtools-safe-area-top: env(s
afe-area-inset-top, 0px); --nuxt-devtools-safe-area-right:
env(safe-area-inset-right, 0px); --nuxt-devtools-safe-area-
bottom: env(safe-area-inset-bottom, 0px); --nuxt-devtools-s
afe-area-left: env(safe-area-inset-left, 0px);">
··· ▼ <head> == $0
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, init
    ial-scale=1">
    <title>My Nuxt.js project</title>
    <link rel="modulepreload" as="script" crossorigin href=
    "/_nuxt/C:/Users/Gebruiker/Desktop/nuxt-project/node_mo
    dules/nuxt/dist/app/entry.js">
    <meta name="description" content="My Nuxt.js project">
    <meta name="keywords" content="Nuxt.js, Tailwind, CSS,
    JavaScript">
    <link rel="icon" type="image/x-icon" href="/favicon.ic
    o">
    <link rel="https://fonts.googleapis.com/css2?family=Mat
    erial+Icons">
```

2. Using useHead()

- The useHead() composable function allows you to manage your <head> tags *programmatically* and *reactively*.
- Can only be used in `<script setup>` blocks
- Page-specific overrides for `nuxt.config.ts`
- You have access to variables (which are defined on the component)
- For instance, if you want the page title to be a specific `Id` or product or description, or anything:

```
// [id].vue: Use the post Title as the page title
useHead({
  title: 'Post details : ' + (post ? post['title'] : id),
  meta: [
    {
      name: 'description',
      content: (post ? post['body'] : 'No post found')
    }
  ]
})
```

Results



Post id: #1

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

Message:

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut

```
<!DOCTYPE html>
<html data-capo style="--nuxt-devtools-safe-area-to
afe-area-inset-top, 0px); --nuxt-devtools-safe-area
env(safe-area-inset-right, 0px); --nuxt-devtools-sa
bottom: env(safe-area-inset-bottom, 0px); --nuxt-de
afe-area-left: env(safe-area-inset-left, 0px);">
 ▼<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-wid
    ial-scale=1">
···  ▼<title> == $0
      "Post details : sunt aut facere repellat provi
      occaecati excepturi optio reprehenderit"
    </title>
    <link rel="modulepreload" as="script" crossorig
    "/_nuxt/C:/Users/Gebruiker/Desktop/nuxt-project
    dules/nuxt/dist/app/entry.js">
```

(further below you find the tag `<meta name="description" … />`)

2a. Using useSeoMeta()

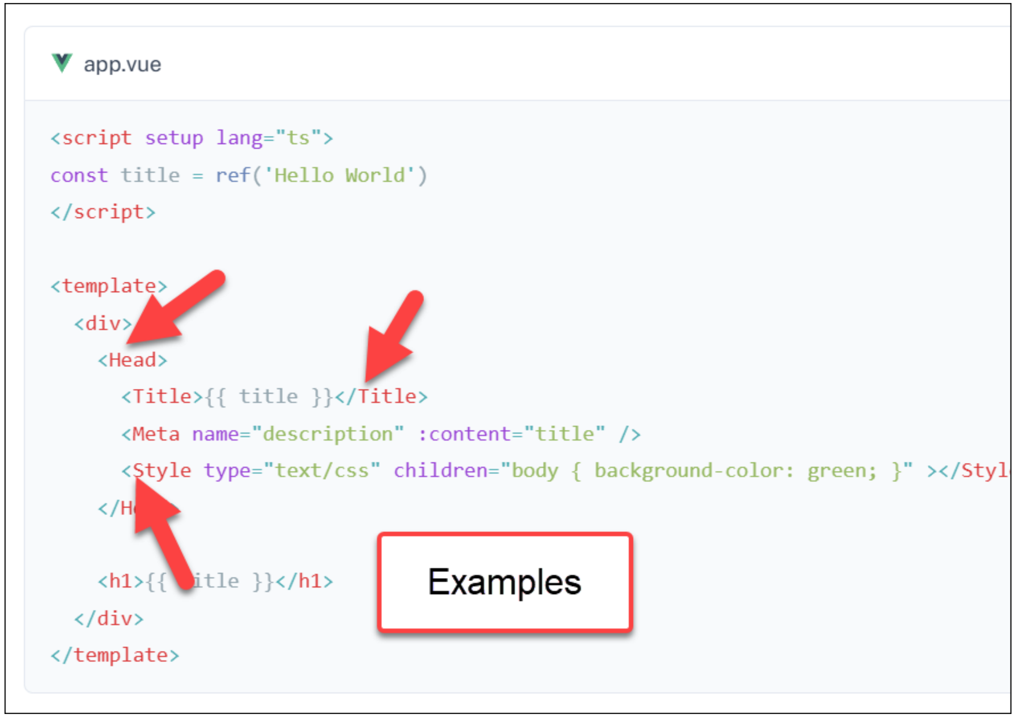
- *Nuxt-specific composable* for SEO Open Graph Metatags for Facebook, X, Instagram and other social platforms, using the ogXXX tags
- Use it like for instance:

```
useSeoMeta({
  title: 'Post details :' + (post.title ??  id),
  ogTitle: 'Using post title as ogTitle...',
  description: 'Description of the page/post',
  ogDescription: 'Description that is used on the og:description meta tag',
  ogImage: 'https://example.com/image.png',
  twitterCard: 'summary_large_image',
})
```

More info, for instance: https://ahrefs.com/blog/open-graph-meta-tags/

3. Using specific Nuxt Components

- Nuxt provides `<Title>`, `<Base>`, `<NoScript>`, `<Style>`, `<Meta>`, `<Link>`, `<Body>`, `<Html>` and `<Head>` components
- This way you can interact directly with metadata within the component's template.
- Because these component names match native HTML elements, they must be capitalized in the template.
- Used less often
- Use them if you prefer to work in template instead of script.
- They override settings in `nuxt.config.ts`



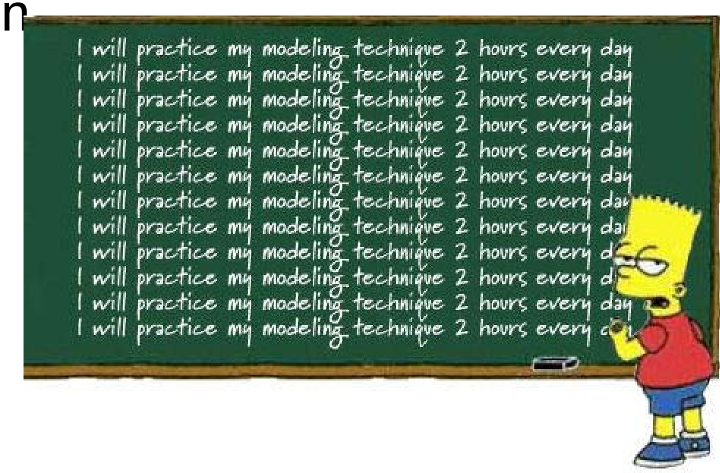
app.vue

```
<script setup lang="ts">
const title = ref('Hello World')
</script>

<template>
  <div>
    <Head>
      <Title>{{ title }}</Title>
      <Meta name="description" :content="title" />
      <Style type="text/css" children="body { background-color: green; }" ></Styl
    </Head>

    <h1>{{ title }}</h1>
  </div>
</template>
```

Examples

Workshop

- Create custom SEO-tags for:
  - One of the static pages in your application
  - The page with the list of products | customers | data, etc.
  - Dynamic SEO-tags for the detail component
- Study the Open Graph tags, experiment with adding them to detail pages.
  - The og:-tags show up if you post them to Facebook, X, and so on.
  - See official documentation at https://ogp.me/
- Read the documentation on SEO and Meta tags on https://nuxt.com/docs/getting-started/seo-meta
- Example: `../examples/170-Seo-Meta`



Checkpoint

- You know three ways of adding page-specific SEO and Meta data:
  - Via `nuxt.config.ts`
  - Via `useHead()`
  - Via Nuxt Specific components like `<Head>`, `<Title>` and more.
- Bonus: you know how to add Open Graph Metatags to your page(s)