NUXT.JS ESSENTIALS CHEAT SHEET

Vue Mastery

STARTING A NEW PROJECT

From Nuxt toolkit:

```
$ npx create-nuxt-app <project-name>
$ cd <project-name>
$ npm install    Installs dependencies
$ npm run dev    Launches the project
```

From scratch:

package.json

```
{
  "name": "my-app",
  "scripts": {
    "dev": "nuxt"
  }
}
```

```
$ npm install --save nuxt
  Installs nuxt and saves it in package.json
$ npm run dev
  Runs the nuxt script / initiates the app
```

FOLDER STRUCTURE

- **ASSETS** - Uncompiled assets (like Less / Sass).
- **STATIC** - Unchanging files (like robots.txt).
- **COMPONENTS**
- **LAYOUTS** - Application layouts.
- **MIDDLEWARE** - Custom functions which run before pages.
- **PAGES** - Application views & routes from which the router is dynamically generated.
- **PLUGINS** - JS plugins run before Vue.js init.
- **STORE** - Vuex Store files.

PAGES

Nuxt reads the file tree inside the `pages` directory to create your application's routes:

```
pages
  index.vue    loaded when root path /
  users
    index.vue    /users path
    _id.vue    _ defines a dynamic route with a param /users/123
```

PAGE KEYS

```
export default {
  asyncData (context) {
    return
axios.get(`https://my-api/posts/${context.params.id}`)
.then((res) => {
      return { title: res.data.title }
    })
  },
  fetch (context) {
    return axios.get('http://my-api/stars').then((res)
=> {
      context.store.commit('setStars', res.data)
    })
  },
  head () {    Set the HTML Head tags for the current page. Uses vue-meta.
    return {
      title: this.title,    Your component's data is available with this.
      meta: [
        { hid: 'description', name: 'description',
          content: 'My custom description' }
      ]
    }
  },
  layout: 'my-custom-layout',    Choose a custom layout for your page.
  validate (context) {    If false, Nuxt loads the error page instead.
    return /^\\d+$/.test(context.params.id)    Must be a number
  },
  transition: {    Define a custom transition for current page
    name: 'my-custom-transition',
    mode: 'out-in'
  }
}
```

Ketabton.com

This cheat sheet is created by
VueMastery.com
The ultimate learning resource for Vue developers.
Need help on your path to Vue Mastery? Check out our tutorials.

(c) ketabton.com: The Digital Library

NUXT.JS ESSENTIALS CHEAT SHEET

Vue Mastery

NUXT COMPONENTS

Use `<nuxt-link>` to navigate between pages in your components.

```
<nuxt-link v-bind:to="'/users' + user.name" >
{{ user.fullName }}'s Profile</nuxt-link>
```

Use `<nuxt-child/>` to display child component routes in your nested routes.

```
<template>
  <div>
    <h1>I am the parent view</h1>
    <nuxt-child />
  </div>
</template>
```

LAYOUTS

Put application layouts in the `layouts` folder. Use the `nuxt` component to display content.

layouts/my-custom-layout.vue

```
<template>
  <div class="container">
    <nuxt />
  </div>
</template>
```

ERROR PAGES

Customize the error page with `layouts/error.vue`. Do not include `<nuxt/>` inside its template.

```
<template>
  <h1 v-if="error.statusCode === 404">
     Page not found
  </h1>
  <h1 v-else>An error occurred</h1>
  <nuxt-link to="/">Home page</nuxt-link>
</template>

<script>
export default {
  props: ['error'],
  layout: 'my-error-layout'
}
</script>
```

You can set a custom layout for the error page

ALIASES FOR REFERENCING FILES

`~` to reference the source directory

```
<template>
  <img src="~/assets/your_image.png"/>
</template>
<script>
import Visits from '~/components/Visits'
export default {
  components: { Visits }
}
</script>
```

VUEX STORE

Nuxt automatically adds Vuex to your project if you have an `index.js` file in your `store` folder. This file must export a method which returns a Vuex instance.

You can now use `this.$store` inside of your components.

```
<template>
  <button @click="$store.commit('increment')">
    {{ $store.state.counter }}
  </button>
</template>
```

ADDITIONAL VUE LEARNING

VueMastery.com

Dive deep into advanced Vue concepts in our **Advanced Components** course.

- Learn the full functionality of Vue
- Understand how Vue internals work together
- Discover how to extend Vue as needed

Plus **5** bonus videos exploring the Vue source code with Evan You, the creator of Vue.

(c) ketabton.com: The Digital Library

NUXT.JS ESSENTIALS CHEAT SHEET

Vue Mastery

NUXT.CONFIG.JS

For configuring your application

```
export default {
  css: [  To add global CSS files
        'bulma/css/bulma.css',
        '~/css/main.css'
      ],
  generate: {  To generate static pages from dynamic routes, specify them here
    routes: function () {
      return [
        '/users/1',
        '/users/2',
        '/users/3'
        ];
    }
  },
  loading: '~/components/loading.vue',
  head: {  Set a custom loading component
    meta: [  Set HTML Head tags for every page
      { charset: 'utf-8' },
      { name: 'viewport', content:
        'width=device-width, initial-scale=1' }
    ],
    link: [{
      rel: 'stylesheet',
      href: 'https://font.com',
    }]
  },
  transition: {  Set the default transition for all pages
    name: 'page',
    mode: 'out-in'
  },
  plugins: ['~/plugins/vue-notifications']
}
```

NUXT

Nuxt.js is an open source project supported by the community. To sponsor the project, head over to Open Collective: opencollective.com/nuxtjs

DEPLOYMENT METHODS

1. SPA (SINGLE PAGE APPLICATION)

Benefit: Organize your project using convention over configuration folder structure and config files. Easy development server.

- Change `mode` in `nuxt.config.js` to `spa`.
- Run `npm run build`
- Deploy the created dist/ folder to your static hosting like GitHub Pages for example.

2. STATIC

Benefit: All pages get pre-rendered into HTML and have a high SEO and page load score. The content is generated at build time.

- Run `npm run generate`
- Deploy the created `dist/` folder with all the generated HTML files and folders to your static hosting.

3. UNIVERSAL

Benefit: Execute your JavaScript on both the client and the server. All routes have high SEO and page load score. Dynamically get routes rendered on the server before being sent to client.

- Upload the contents of your application to your server of choice.
- Run `nuxt build` to build your application.
- Run `nuxt start` to start your application and start accepting requests.

RUNNING AND BUILDING

```
$ nuxt
Launch a development server on Localhost:3000 with hot-reloading
$ nuxt generate
Build the application and generate every route as a HTML file
$ nuxt build
Build your application for production with webpack and minify assets
$ nuxt start
Start the server in production mode
```

Need help on your path to Vue Mastery? Check out our tutorials on **VueMastery.com**

Get more e-books from www.ketabton.com
Ketabton.com: The Digital Library