

Structure

Directories

layouts

앱의 기본 골격을 생성

default.vue 가 기본

```
default.vue

<template>
  <div>
    <nuxt />
  </div>
</template>
```

`<nuxt />` 앱의 시작점

페이지별로 다른 레이아웃을 사용 가능

```
layouts/intro.vue

<template>
  <div>
    <intro-header></intro-header>
    <nuxt />
    <intro-footer></intro-footer>
  </div>
</template>

pages/intro/index.vue
<template>
  <div>
    intro
  </div>
</template>
<script>
export default {
  layout: 'intro',
}
</script>
```

사용자 오류 페이지도 layouts에 생성

```
error.vue

<template>
  <div>
    <h1 v-if="error.statusCode === 404">page not found</h1>
    <h1 v-else>something wrong...</h1>
  </div>
</template>
```



사용자 오류 페이지는 layouts에 위치하지만 page로 취급되어야 합니다. pages의 뷰에서 layout으로 지정하면 안됨~!

pages

layouts에 다이내믹 라우팅되어 페이지를 생성

pages디렉토리의 하위 디렉토리별로 page컴포넌트를 생성하면 자동으로 routes에 정리된다.

필수 디렉토리

nuxt.config.js에서 디렉토리명 변경 가능

```
pages/intro/index.vue

<template></template>
<script>
export default {
  asyncData() {},
  fetch() {},
  head() {},
  ...
}
</script>
```

Pages methods

components

컨텐츠를 구성하는 컴포넌트들이 위치함

assets

css, image 요소 등을 관리하기 위해 생성되는 기본 디렉토리. webpack vue loader 처리된다. 필수 디렉토리는 아니며 nuxt.config.js 에서 이름 바꿀 수 있다.

middleware

페이지가 랜더링 되기 이전에 실행할 함수 등을 모아놓는 디렉토리. 필수 디렉토리는 아니며 nuxt.config.js에서 이름 바꿀 수 있다.

static

정적 파일들을 위한 디렉토리.

store

Vuex store 파일들이 위치하는 디렉토리. index.js 파일을 생성시킴으로써 자동으로 활성화된다.

plugins

애플리케이션이 실행되기 이전에 불러올 js 플러그인을 포함한다.