

link: null  
title: 珠峰架构师成长计划  
description: Nuxt.js是使用 Webpack 和 Node.js 进行封装的基于Vue的SSR框架  
keywords: null  
author: null  
date: null  
publisher: 珠峰架构师成长计划  
stats: paragraph=40 sentences=39, words=290

## 什么是nuxt #

Nuxt.js是使用 Webpack 和 Node.js 进行封装的基于Vue的SSR框架

## nuxt特点 #

优点:  
更好的 SEO, 由于搜索引擎爬虫抓取工具可以直接查看完全渲染的页面。首屏渲染速度快

缺点:  
Node.js 中渲染完整的应用程序, 显然会比仅仅提供静态文件的 **server** 更加大量占用 CPU 资源。需要考虑服务器负载, 缓存策略

## 项目生成 #

```
npx create-nuxt-app nuxt-project  
cd nuxt-project  
yarn dev
```

## 项目目录 #

- **assets** 静态资源 会被webpack处理
- **static** 不会被webpack处理
- **components** 公共组件
- **layout**布局组件
- **pages**路由页面 可以生成对应的路由
- **middleware** 运行过程中发生的事
- **store** 存放vuex
- **plugins** 存放javascript插件的
- **nuxt.config.js** 存放nuxt配置文件
- 别名默认可以采用 ~ 或者 @符号

## nuxt.config.js配置 #

- **env** 可以配置环境变量通过cross-env

```
env: {  
  baseUrl: process.env.BASE_URL  
}
```

- **cache:false** // 提升组件缓存策略
- **css** 全局css样式
- **head** 配置头
- **loading** (需要等待\$loading 挂载完成)

```
loading: { color: '#000',height:'10px' }  
mounted() {  
  this.$nextTick(()=>{  
    this.$nuxt.$loading.start()  
  });  
}
```

- **modules** 存放第三方模块 @nuxtjs/axios 第三方模块
- **plugins** 配置插件
- **transition**动画效果

```
config.js  
transition: {  
  name: 'layout',  
  mode: 'out-in'  
},  
.layout-enter-active, .layout-leave-active {  
  transition: opacity .5s  
}  
.layout-enter, .layout-leave-active {  
  opacity: 0  
}
```

使用history.pushState跳转页面,不会触发页面整体重新渲染

- 路径参数

```
<nuxt-link to="/user/4/5">{{#x8DEF; &#x5FB4; &#x53C2; &#x6570; </nuxt-link>
```

- 查询字符串
- **validate**方法 必须返回是否访问这个页面, 返回false执行404逻辑

```
export default {  
  validate({params}) {  
    return params.id !== 4;  
  }  
}
```

- 动画效果

```
.page-enter-active, .page-leave-active {  
  transition: opacity .5s;  
}  
.page-enter, .page-leave-active {  
  opacity: 0;  
}
```

## 中间件 #

如果经过服务端则在服务端执行

- 全局中间件
- layout 在layout中增加middleware
- 组件中间件 在组件中增加middleware

```
export default function ({ store, redirect }) {
  if (!store.state.user) {
    return redirect('/login')
  }
}
router: {
  middleware: 'auth'
}
```

## layout配置 #

- 自定义error页面 增加error.vue(配置错误layout)
- 自定义layout布局

增加错误页面，错误页需要配置layout

```
export default {
  props: ['error'],
  layout: 'page'
}
```

## 数据获取 #

- asyncData使用(仅在页面组件中使用)

```
async asyncData ({ params }) { // 65E0;this
  let { data } = await axios.get();
  return { title: data }
}
```

## 插件的使用 #

扩展原型上的方法plugins

```
export default function ({ app }, inject) {
  inject('my', () => { // 5728;app64E0A;6548C;this690FD;66CE8;6518C;68FD9;64E2A;665B9;66CD5;
    console.log('my');
  })
}
```

## 使用vuex #

## 运行流程 #

nextServerninit 只在主模块中使用  
nuxt.config.js 全局中间件  
matching layout 不同布局的中间件  
matching page & children 页面中间件  
validate 返回false显示错误页面  
asyncData 服务端渲染的页面数据请求  
fetch 同步vuex数据

## 登录实战 #

element-ui + redis + mongo + nuxt

<https://github.com/zhufengzhufeng/nuxt-login.git> (<https://github.com/zhufengzhufeng/nuxt-login.git>)