

Nuxt 作业

作业要求

截止日期为2021年8月1日23: 59.

本次作业为你提供了一个框架，你需要在这个框架之上实现一个简单的博客页面。

该博客功能如下：

- 提供主页、"关于"页面和博文页面
- 主页边栏可以查看北京天气
- 主页点击标签可以进行筛选

你要完成的任务如下。每题各问不分先后，你可以先完成简答题再完成代码填空。
满分100分，此外有10分附加分。

你需要在报告文件 `report.pdf` 内写明邮箱、姓名、班级、学号，然后提交。

提交方法如下：

- 1 | 学号_班级_姓名
- 2 | `___ code` ... 此文件夹内放置提供的工程，使得 ``packages.json`` 直接位于其下。
- 3 | `___ report.pdf` ... 作业简答题的PDF文件，按照题号逐题给出。只接受PDF。
- 4 | `___ weather.jpg` ... JPG 格式，主页天气信息组件渲染效果截图。
- 5 | `___ article.jpg` ... JPG 格式，任意文章页面的渲染效果截图。

然后打包上述文件夹为 `学号_班级_姓名.zip` (只接受zip)，发送到 `me@panda2134.site`。

1. 修改 `components/WeatherInfo.vue`。(40分)

- (30分) 编程：对这个组件进行填空，使得其能从下文提到的天气 API 获得北京天气信息，并且显示位置、天气图标、温度和天气状况。
在文件中已经给出了部分代码用于 API 调用加载中的显示占位。
你可以参考下面的[文档](#)
- (3分) 解释：`vue-content-placeholders` 组件挂载和 `async fetch()` 成功获得 API 数据的先后关系如何？可参考：
[Nuxt 2.12 中 fetch 如何工作](#)
- (5分) 解释：axios 是什么？通过 `this.$axios` [使用它](#) 比使用 `XMLHttpRequest` 有何好处？
相比 [Fetch API](#) 呢？
- (2分) 解释：在加载时为什么放置 `vue-content-placeholder`？
用 `v-if` 判断未加载完什么都不显示不行吗？哪种方法更好？

2. 修改 `pages/article/_id.vue`。(30分+10分)

- 这个页面的逻辑是利用 Nuxt Content 模块读取 `content/article` 目录下的博文内容并且显示。
提示：`this.$route.params.id` 存放着参数 `id`。你可以参考 `pages/about.vue` 进行实现。
- (5分) 编程：利用 [validate](#) 接口，实现对参数 `id` 的检验；如果 `content/article` 下没有文件名等于 `id`（不考虑扩展名）的文件，则返回 `false` 说明页面不存在，此时 HTTP 状态码自动设为 404。
- (15分) 编程：在 `async fetch()` 中补充读取当前文章并显示的实现。

同时，在 Vuex Store 中设置页面标题和副标题，以更新 `layouts/default.vue` 中对应的显示。

- (3分) 解释：Vuex Store 中 Mutation 和 Action 的区别是什么？
- (5分) 解释：`this.$content` 是怎么用来读取文章内容的？可以参考 [Fetching Content](#) 进行解释；在 Nuxt 组件中执行 `this.$content('article').fetch()` 返回值是什么？为什么要用 `<nuxt-content>` 组件进行显示？
- (2分) 解释：`middleware/title.js` 作用是什么？如果有页面设置了 Vuex 中标题为特定值，跳转到下个路由页面后这个新标题会保留吗？
- (附加题，10分) 解释：我是如何把 Nuxt Content 自带的 [PrismJS](#) 高亮替换为 [Shiki](#) 高亮器的？阅读 `nuxt.config.js` 和 `utils/highlighter.js` 解释之。

3. 阅读主页代码 `pages/index.vue`。(20分)

- 解释 (5分)：主页中如何统计的所有标签？可以参考 [lodash 文档](#)。
- 解释 (15分)：主页中如何实现的按照标签筛选？
 - (5分) 为什么使用 `computed` 属性，而不是 `methods` 呢？
 - (10分) 为什么要采用 `decodeURIComponent` 呢？

天气 API 简单介绍如下。

- API 请求为向 `https://api.weatherapi.com/v1/current.json` 这一 URL 发送的 GET 请求。
 - GET 请求的请求参数 (Query Params) 如下：
 - `aqi` 设为 `no`：不请求空气质量信息
 - `q` 设为 `props` 中的 `city` (即 `this.city`)：北京天气
 - `key` 设为 `this.$config.WEATHER_API_KEY`：采用 `nuxt.config.js` 提供的 API 密钥。
 - 关于 `this.$config` 可以参考 [文档](#)
 - 关于请求发送，可以采用 Axios，参考已经写好的部分代码。
 - 返回值为 `application/json` 格式文档。设其为对象 `currentWeather`。
 - `currentWeather.current.condition.icon` 存放着天气图标的 URL；
 - `currentWeather.current.condition.text` 为当前天气信息；
 - `currentWeather.current['temp_c']` 为摄氏温度
 - `currentWeather.location.name` 为地理位置
- 样例如下。

```
1 {
2   "location": {
3     "name": "Beijing",
4     "region": "Beijing",
5     "country": "China",
6     "lat": 39.93,
7     "lon": 116.39,
8     "tz_id": "Asia/Shanghai",
9     "localtime_epoch": 1626772135,
10    "localtime": "2021-07-20 17:08"
11  },
12  "current": {
```

```
13     "last_updated_epoch": 1626768000,  
14     "last_updated": "2021-07-20 16:00",  
15     "temp_c": 31.0,  
16     "temp_f": 87.8,  
17     "is_day": 1,  
18     "condition": {  
19         "text": "Sunny",  
20         "icon": "//cdn.weatherapi.com/weather/64x64/day/113.png",  
21         "code": 1000  
22     },  
23     "wind_mph": 4.3,  
24     "wind_kph": 6.8,  
25     "wind_degree": 150,  
26     "wind_dir": "SSE",  
27     "pressure_mb": 1010.0,  
28     "pressure_in": 30.3,  
29     "precip_mm": 0.0,  
30     "precip_in": 0.0,  
31     "humidity": 63,  
32     "cloud": 0,  
33     "feelslike_c": 34.3,  
34     "feelslike_f": 93.7,  
35     "vis_km": 10.0,  
36     "vis_miles": 6.0,  
37     "uv": 7.0,  
38     "gust_mph": 7.4,  
39     "gust_kph": 11.9  
40 }  
41 }
```

构建安装

```
1  # install dependencies  
2  $ yarn install  
3  
4  # serve with hot reload at localhost:3000  
5  $ yarn dev  
6  
7  # build for production and launch server  
8  $ yarn build  
9  $ yarn start  
10  
11 # generate static project  
12 $ yarn generate
```

参阅 [文档](#) 以获取更多信息。

特殊目录

你可以创建以下的额外目录，其中某些有特殊行为。只有 `pages` 是必须的。若你不使用对应功能，你可以删除它们。

assets

The assets directory contains your uncompiled assets such as Stylus or Sass files, images, or fonts.

More information about the usage of this directory in [the documentation](#).

components

The components directory contains your Vue.js components. Components make up the different parts of your page and can be reused and imported into your pages, layouts and even other components.

More information about the usage of this directory in [the documentation](#).

layouts

Layouts are a great help when you want to change the look and feel of your Nuxt app, whether you want to include a sidebar or have distinct layouts for mobile and desktop.

More information about the usage of this directory in [the documentation](#).

pages

This directory contains your application views and routes. Nuxt will read all the `*.vue` files inside this directory and setup Vue Router automatically.

More information about the usage of this directory in [the documentation](#).

plugins

The plugins directory contains JavaScript plugins that you want to run before instantiating the root Vue.js Application. This is the place to add Vue plugins and to inject functions or constants. Every time you need to use `vue.use()`, you should create a file in `plugins/` and add its path to plugins in `nuxt.config.js`.

More information about the usage of this directory in [the documentation](#).

static

This directory contains your static files. Each file inside this directory is mapped to `/`.

Example: `/static/robots.txt` is mapped as `/robots.txt`.

More information about the usage of this directory in [the documentation](#).

store

This directory contains your Vuex store files. Creating a file in this directory automatically activates Vuex.

More information about the usage of this directory in [the documentation](#).