

# 我的博客vue2

---

设定添加博客并实现post数据

App.vue中：

AddBlog.vue中绑定数据：

添加博客页面的样式

展示博客并请求数据

自定义指令（钩子函数）

展示博客页面的搜索功能

vue路由

路由参数

提交数据信息

博客的编辑和删除功能

axios替换vue-resource

配置axios

## 设定添加博客并实现post数据

App.vue中：

1. 引入模块

```
1  import AddBlog from "../components/AddBlog";
```

2. 引入组件

```
1 export default {  
2   name: "App",  
3   components: {  
4     AddBlog,  
5   },  
6 };
```

### 3. 使用组件

```
1 <add-blog></add-blog>  
2
```

```
<template>  
  <div id="app">  
    <add-blog></add-blog>  
  </div>  
</template>  
  
<script>  
import AddBlog from "../components/AddBlog";  
  
export default {  
  name: "App",  
  components: {  
    AddBlog,  
  },  
};  
</script>
```

## AddBlog.vue中绑定数据：

### 1. 获取输入框中的值：

```

<template>
  <div class="add-blog">
    <h2>添加博客</h2>
    <form>
      <label>博客标题</label>
      <input type="text" v-model="blog.title" required />
      <label>博客内容</label>
      <textarea v-model="blog.content"> />
    </form>
  </div>
</template>

<script>
export default {
  name: "add-blog",
  data() {
    return {
      blog: {
        title: "",
        content: "",
      },
    };
  },
};
</script>

```

**绑定数据**

## 2. 展示输入框中的值

```

<div id="preview">
  <h3>博客总览</h3>
  <p>博客标题: {{ blog.title }}</p>
  <p>博客内容: </p>
  <p>{{ blog.content }}</p>
</div>

```

**展示数据**

## 3. 绑定选项

```

<div id="checkboxes">
  <label>vue.js</label>
  <input type="checkbox" value="vue.js" v-model="blog.categories" />
  <label>node.js</label>
  <input type="checkbox" value="node.js" v-model="blog.categories" />
  <label>react.js</label>
  <input type="checkbox" value="react.js" v-model="blog.categories" />
  <label>angular</label>
  <input type="checkbox" value="angular" v-model="blog.categories" />
</div>

```

**数据绑定**

## 4. 创建选项数组

```

return {
  blog: {
    title: "",
    content: "",
    categories: [],
  },
};

```

**创建选项数组**

## 5. 遍历拿取数组中的值并展示

```
<p>博客分类: </p>
<ul>
  <li v-for="category in blog.categories">{{ category }}</li>
</ul>
```

## 6. 创建作者数组

```
data() {
  return {
    blog: {
      title: "",
      content: "",
      categories: [],
    },
    authors: ["小明", "小红", "小张", "小李"],
  };
};
```

创建作者数组

## 7. 遍历得到作者数组选项

```
<label>作者: </label>
<select>
  <option v-for="author in authors">{{ author }}</option>
</select>
```

遍历得到作者数组

## 8. 绑定作者

```
<label>作者: </label>
<select v-model="blog.author">
  <option v-for="author in authors">{{ author }}</option>
</select>
</form>
```

绑定作者

## 9. 创建博客作者

```
data() {
  return {
    blog: {
      title: "",
      content: "",
      categories: [],
      author: "",
    },
    authors: ["小明", "小红", "小张", "小李"],
  };
};
```

创建博客作者空数组

## 10. 创建提交按钮

绑定点击事件并阻止事件默认行为（刷新页面）

添加方法名: post

```
<button v-on:click.prevent="post">添加博客</button>
```

## 11. 下载资源

```
npm install vue-resource --save
```

## 12. 引入模块

在main.js中:

```

main.js > ...
// The Vue build version to load with the
// (runtime-only or standalone) has been
import Vue from 'vue'
import VueResource from 'vue-resource'
import App from './App'
import router from './router'

Vue.config.productionTip = false
Vue.use(VueResource)

/* eslint-disable no-new */
new Vue({
  el: '#app',
  router,
  components: { App },
  template: '<App/>'
})

```

引入

使用

### 13. 实现post方法:

post 网址

<https://jsonplaceholder.typicode.com/>

<https://jsonplaceholder.typicode.com/posts>

JSON-handle

JSON :

```

0
  ...
  userId: 1
  id: 1
  title: sunt aut facere repellat provident occaecati excepturi optio reprehenderit
  body: quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem su
1
  ...
  userId: 1
  id: 2
  title: qui est esse
  body: est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel
2
  ...
  userId: 1
  id: 3
  title: ea molestias quasi exercitationem repellat qui ipsa sit aut
  body: et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius
3
  ...
  userId: 1
  id: 4
  title: eum et est occaecati
  body: ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur dc

```

```

methods: {
  post: function () {
    this.$http
      .post("https://jsonplaceholder.typicode.com/posts", {
        title: this.blog.title,
        body: this.blog.content,
        userId: 1,
      })
      .then(function (data) {
        console.log(data);
      });
  },
}

```

测试数据提交接口

检查是否能够成功提交得到返回的数据

#### 14. 判断是否提交

```

author: ,
],
authors: ["小明", "小红", "小张", "小李"],
submitted:false
};

```

判断是否提交

```

<h2>添加博客</h2>
<form v-if="!submitted">
  <label>博客标题</label>

```

判断是否展示

```

methods: {
  post: function () {
    this.$http
      .post("https://jsonplaceholder.typicode.com/posts", {
        title: this.blog.title,
        body: this.blog.content,
        userId: 1,
      })
      .then(function (data) {
        console.log(data);
        this.submitted = true;
      });
  },
}

```

```

<div v-if="submitted">
  <h3>您的博客发布成功! </h3>
</div>

```

隐藏表单的时候展示

## 添加博客页面的样式

```
1  ▾ #add-blog * {
2      box-sizing: border-box;
3  }
4  ▾ #add-blog {
5      margin: 20px auto;
6      max-width: 600px;
7      padding: 20px;
8  }
9
10 ▾ label {
11     display: block;
12     margin: 20px 0 10px;
13 }
14 input[type="text"],
15 textarea,
16 ▾ select {
17     display: block;
18     width: 100%;
19     padding: 8px;
20 }
21 ▾ textarea {
22     height: 200px;
23 }
24 ▾ #checkboxes label {
25     display: inline-block;
26     margin-top: 0;
27 }
28 ▾ #checkboxes input {
29     display: inline-block;
30     margin-right: 10px;
31 }
32 ▾ button {
33     display: block;
34     margin: 20px 0;
35     background: crimson;
36     border: none;
37     color: #fff;
38     padding: 14px;
39     border-radius: 4px;
40     font-size: 18px;
41     cursor: pointer;
42 }
43
44 ▾ #preview {
45     padding: 10px 20px;
```

```

46     border: 1px dotted #ccc;
47     margin: 30px 0;
48   }
49   h3 {
50     margin-top: 10px;
51   }

```

## 展示博客并请求数据

### 1. 新建组件ShowBlog.vue

```

<template>
  <div id="show-blogs"></div>
</template>

<script>
export default {
  name: "show-blogs",
};
</script>

<style>
#app {
}
</style>

```

### 2.

### 3. 在App.vue中引入新组件

```

App.vue | App.vue | template | div#app | show-blog
1  <template>
2    <div id="app">
3      <add-blog></add-blog>
4      <show-blog></show-blog> 3、使用
5    </div>
6  </template>
7
8  <script>
9    import AddBlog from "../components/AddBlog";
10   import ShowBlog from "../components/ShowBlog"; 1、引入
11   export default {
12     name: "App",
13     components: {
14       AddBlog,
15       ShowBlog, 2、组件
16     },
17   };
18 </script>
19
20 <style>
21 #app {
22 }
23 </style>
24

```

### 3. 请求数据



```

created() {
  // 请求数据
  this.$http
    .get("https://jsonplaceholder.typicode.com/posts")
    .then(function (data) {
      console.log(data);
      // 展示10个数据
      data.body.slice(0, 10);
    });
  // 请求数据
}

```

#### 4. 接收数据

```

data() {
  return {
    blogs: [], // 1.定义空数组
  };
},
created() {
  // 请求数据
  this.$http
    .get("https://jsonplaceholder.typicode.com/posts")
    .then(function (data) {
      console.log(data);
      // 展示10个数据
      this.blogs = data.body.slice(0, 10); // 2.接收返回的数据
    });
}

```

1.定义空数组

2.接收返回的数据

#### 5. 调用数据

```

<div id="show-blogs">
  <h1>博客总览</h1>
  <div class="single-blog" v-for="blog in blogs">
    <h2>{{ blog.title }}</h2>
    <article>
      {{ blog.body }}
    </article>
  </div>
</div>
</template>

```

样式：

```
1 ▾ #show-blogs {
2     max-width: 800px;
3     margin: 0 auto;
4 }
5 ▾ .single-blog {
6     padding: 20px;
7     margin: 20px 0;
8     box-sizing: border-box;
9     background: #eee;
10    border: 1px dotted #aaa;
11 }
12 ▾ #show-blogs a {
13     color: #444;
14     text-decoration: none;
15 }
16 ▾ input[type="text"] {
17     padding: 8px;
18     width: 100%;
19     box-sizing: border-box;
20 ▾ }#show-blogs {
21     max-width: 800px;
22     margin: 0 auto;
23 }
24 ▾ .single-blog {
25     padding: 20px;
26     margin: 20px 0;
27     box-sizing: border-box;
28     background: #eee;
29 }
```

## 自定义指令（钩子函数）

### 1. 使用自定义指令

```
<h2 v-rainbow>{{ blog.title }}></h2>
```

### 2. 在main.js中自定义指令

```
Vue.config.productionTip = false
Vue.use(VueResource)

//自定义指令
Vue.directive('rainbow', {
  bind(el, binding, vnode) {
    el.style.color = "#" + Math.random().toString(16).slice(2, 8);
  }
})
```

生成彩虹颜色

### 3. 自定义属性传参数

```
@typ
@
@
@
./src/main.js
<div id="show-blogs" v-theme="wide">
  <h1>博客总览</h1>
  <div class="single-blog" v-for="blog in blogs">
    <div class="blog">
      <h2>{{blog.title}}</h2>
      <p>{{blog.content}}</p>
    </div>
  </div>
</div>
```

[Vue warn]: Property or method "wide" is not defined on the instance vue.esm.js?efeb:628 but referenced during render. Make sure that this property is reactive, either in the data option, or for class-based components, by initializing the property. See: <https://vuejs.org/v2/guide/reactivity.html#Declaring-Reactive-Properties>.

Found in

此处的wide缺少引号，传递的是字符串

```
---> <ShowBlogs> at src/components/ShowBlog.vue
      <App> at src/App.vue
      <Root>
```

```
<template>
  <div id="show-blogs" v-theme="'wide'">
    <h1>博客总览</h1>
  </div>
</template>
```

```
Vue.directive('theme', {
  bind(el, binding, vnode) {
    if (binding.value == 'wide') {
      el.style.maxWidth = '1260px';
    } else if (binding.value == 'narrow') {
      el.style.maxWidth = '560px';
    }
  }
})
```

```
<template>
  <div id="show-blogs" v-theme:column="'narrow'">
    <h1>博客总览</h1>
  </div>
</template>
```

```
Vue.directive('theme', {
  bind(el, binding, vnode) {
    if (binding.value == 'wide') {
      el.style.maxWidth = '1260px';
    } else if (binding.value == 'narrow') {
      el.style.maxWidth = '560px';
    }
    if (binding.arg == 'column') {
      el.style.background = '#6677cc';
      el.style.padding = '20px';
    }
  }
})
```

# 展示博客页面的搜索功能

## 1. 将标题全转为大写

```
<h2 v-rainbow>{{ blog.title | to - uppercase }}</h2>
```

转为大写

## 2. main.js自定义过滤器

在main.js中定义的是全局方法 所有组件都可以使用

```
// 自定义过滤器
Vue.filter("to-uppercase", function (value) {
  // value 接收管道的值
  return value.toUpperCase();
})
```

## 3. 控制详情内容

```
<div class="single-blog">
  <div class="blog-header">
    <h2>{{ blog.title | snippet }}</h2>
  </div>
</div>
```

定义过滤器

```
Vue.filter("snippet", function (value) {
  return value.slice(0, 100) + "...";
})
```

切取数据后拼接省略号

## 4. 实现搜索效果

```
<input type="text" placeholder="搜索" v-model="search" />
<div class="single-blog" v-for="blog in blogs">
  <div class="blog-header">
    <h2>{{ blog.title | snippet }}</h2>
  </div>
</div>
```

绑定数据，获取搜索框输入的内容

```
<h1>博客总览</h1>
<input type="text" placeholder="搜索" v-model="search" />
<div class="single-blog" v-for="blog in filteredBlogs">
  <div class="blog-header">
    <h2 v-rainbow>{{ blog.title | touppercase }}</h2>
  </div>
  <div class="blog-body">
    <div class="blog-content">
      <div class="blog-text">
        <p>{{ blog.body | snippet }}</p>
      </div>
    </div>
  </div>
</div>
```

遍历过滤后的数据，触发computed

```

<h1>博客总览</h1>
<input type="text" placeholder="搜索" v-model="search" />
<div class="single-blog" v-for="blog in filteredBlogs">
  <h2 v-rainbow>{{ blog.title | touppercase }}</h2>
  <article>
    {{ blog.body | snippet }}
  </article>
</div>
</div>
</template>

computed: {
  filteredBlogs: function () {
    // 拿到blogs里面的内容
    return this.blogs.filter((blog) => {
      // 查到内容展示内容，没有查到就返回false 匹配标题
      return blog.title.match(this.search);
      // 匹配内容
      // return blog.body;
    });
  },
},
};

```

match() 方法可在字符串内检索指定的值，或找到一个或多个正则表达式的匹配。  
 该方法类似 indexOf() 和 lastIndexOf()，但是它返回指定的值，而不是字符串的位置。

## 5. 定义局部方法

```

//
computed: { ...
},
filters: {
  touppercase: function (value) {
    return value.toUpperCase();
  },
},
};
/script>

```

es6:

```

//
filters: {
  touppercase(value) {
    return value.toUpperCase();
  },
},
};

```

## 6. 局部自定义指令

```

directives: {
  rainbow: {
    bind(el, binding, vnode) {
      el.style.color = "#" + Math.random().toString(16).slice(2, 8);
    },
  },
},
};

```

# vue路由

```
npm install vue-router --save
```

1. 在main.js中引入路由
2. 使用路由

```
import Vue from 'vue'
import VueResource from 'vue-resource'
import VueRouter from 'vue-router' ← 引用
import App from './App'
import router from './router'

Vue.config.productionTip = false
Vue.use(VueResource)
Vue.use(VueRouter) ← 使用
```

3. 创建routes.js

```
// 2.引入对象
import ShowBlogs from './components/ShowBlogs.vue'
import AddBlog from './components/AddBlog.vue'

// 1.返回对象
export default [
  {
    path: '/',
    component: ShowBlogs
  },
  {
    path: '/add',
    component: AddBlog
  }
]
```

4. 在main.js中引入

```
import Vue from 'vue'
import VueResource from 'vue-resource'
import VueRouter from 'vue-router'
import App from './App' ← 引入routes.js
import Routes from './routes'
```

5. 创建路由并使用

```
// 创建路由
const router = new VueRouter({
  routes: Routes
})

/* eslint-disable no-new */
new Vue({
  el: '#app',
  components: { App },
  template: '<App/>',
  router: router
})
```

创建路由并使用

6. 将app.vue中更改为 `<router-view></router-view>`

```
<!-- <add-blog></add-blog> -->
<!-- <show-blog></show-blog> -->
<router-view></router-view>
```

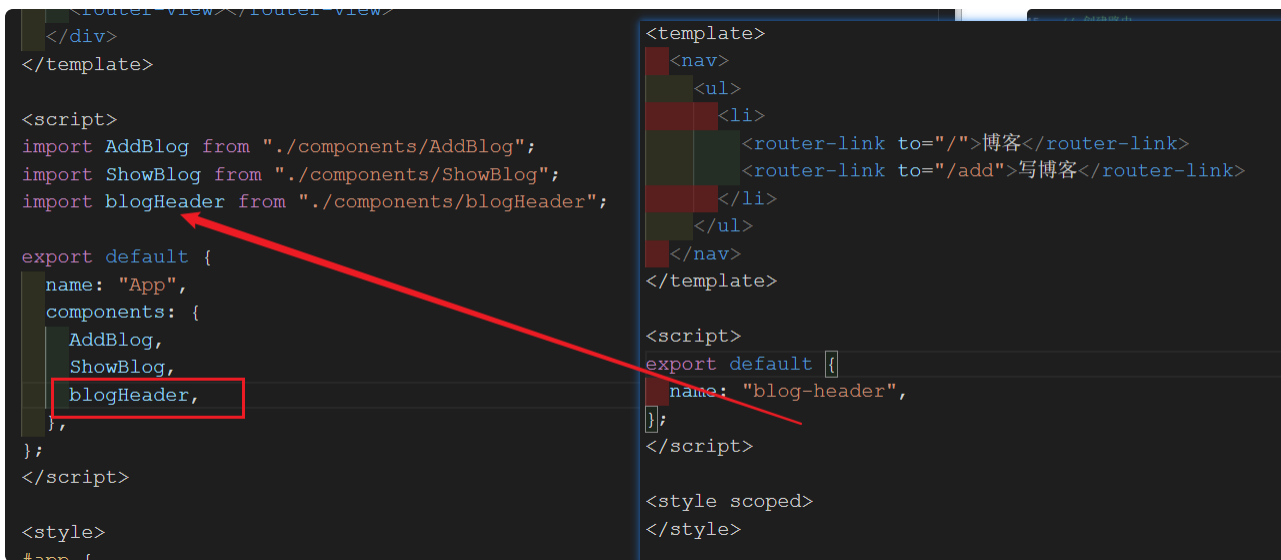
```
45 // 创建路由
46 const router = new VueRouter({
47   routes: Routes,
48   mode: "history"
49 })
50
51
```

去掉路由的#号

7. 新建BlogHeader.vue组件

```
1 <template>
2   <nav>
3     <ul>
4       <li>
5         <router-link to="/">博客</router-link>
6         <router-link to="/add">写博客</router-link>
7       </li>
8     </ul>
9   </nav>
10 </template>
11
12 <script>
13   export default {
14     name: "blog-header",
15   };
16 </script>
17
18 <style scoped>
19 </style>
```

## 8. 在app.vue中引入组件



```
<router-view></router-view>
</div>
</template>

<script>
import AddBlog from "../components/AddBlog";
import ShowBlog from "../components/ShowBlog";
import blogHeader from "../components/blogHeader";

export default {
  name: "App",
  components: {
    AddBlog,
    ShowBlog,
    blogHeader,
  },
};
</script>

<style>
#app {
  </style>
```

```
<template>
<nav>
  <ul>
    <li>
      <router-link to="/">博客</router-link>
      <router-link to="/add">写博客</router-link>
    </li>
  </ul>
</nav>
</template>

<script>
export default {
  name: "blog-header",
};
</script>

<style scoped>
</style>
```

## 9. 使用组件



```
App.vue > {} App.vue > template > div#app > blog-header
<template>
  <div id="app">
    <!-- <add-blog></add-blog> -->
    <!-- <show-blog></show-blog> -->
    <blog-header></blog-header>
    <router-view></router-view>
  </div>
</template>
```

使用组件

## 10. 添加样式

```
<li>
  <router-link to="/" exact>博客</router-link>
  <router-link to="/add" exact>写博客</router-link>
</li>
</ul>
<style scoped>
/ul>
/empl>
  ul {
    list-style-type: none;
    text-align: center;
    margin: 0;
  }
  li {
    display: inline-block;
    margin: 0 10px;
  }
  a {
    color: #fff;
    text-decoration: none;
    padding: 12px;
    border-radius: 5px;
  }
  nav {
    background: crimson;
    padding: 30px 0;
    margin-bottom: 40px;
  }
  .router-link-active {
    background: rgba(255, 255, 255, 0.8);
    color: #444;
  }
}
```

选中样式

# 路由参数

## 1. 创建路由参数

```

// 2.引入对象
import ShowBlogs from './components/ShowBlog.vue'
import AddBlog from './components/AddBlog.vue'
import SingleBlog from './components/SingleBlog.vue'

// 1.返回对象
export default [
  {
    path: '/',
    component: ShowBlogs
  },
  {
    path: '/add',
    component: AddBlog
  },
  {
    path: '/blog/:id',
    component: SingleBlog
  }
]

```



创建路由参数

## 2. 创建singleBlog组件

```

<template>
  <div id="single-blog"></div>
</template>

<script>
export default {
  name: "single-blog",
  data() {
    return;
  },
};
</script>

```

```

components > Singleblog.vue > {} Singleblog.vue > script > default
1 <template>
2   <div id="single-blog">
3     <h1>{{ blog.title }}</h1>
4     <article>{{ blog.body }}</article>
5   </div>
6 </template>
7
8 <script>
9 export default {
10   name: "single-blog",
11   data() {
12     return {
13       id: 4,
14       blog: {},
15     };
16   },
17   created() {
18     this.$http
19       .get("https://jsonplaceholder.typicode.com/posts/" + this.id)
20       .then(function (data) {
21         console.log(data);
22         this.blog = data.body;
23       });
24   },
25 };
26 </script>

```

使用数据

拼接id

存储数据

### 3. 拿到路由id

```

name: "single-blog",
data() {
  return {
    // 拿到路由id
    id: this.$route.params.id,
    blog: {},
  };
},

```

CSS | 复制代码

```

1 #single-blog {
2   max-width: 960px;
3   margin: 0 auto;
4   padding: 20px;
5   background: #eee;
6   border: 1px dotted #aaa;
7 }

```

### 4. 获取blog路由跳转id

```
<div class="single-blog" v-for="blog in filteredBlogs">
  <router-link v-bind:to="'/blog/' + blog.id">
    <h2 v-rainbow>{{ blog.title | toUppercase }}</h2>
  </router-link>
  <article>
    {{ blog.body | snippet }} 获取blog跳转id
  </article>
```

## 提交数据信息

<https://firebase.google.cn/>

1. addBlog.vue修改post地址

```
methods: {
  post: function () {
    this.$http
      .post(["https://blog-ad61e-default-rtdb.firebaseio.com/posts.",
        json", {
          title: this.blog.title,
          body: this.blog.content,
          userId: 1,
        }])
      .then(function (data) {
        console.log(data);
        this.submmited = true;
      });
  },
```

2. 传递对象

```
data() {
  return { 直接传递对象
    blog: {
      title: "",
      content: "",
      categories: [],
      author: "",
    },
    authors: ["小明", "小红", "小张", "小李"],
    submmited: false,
  };
},
methods: {
  post: function () {
    this.$http
      .post(["https://blog-ad61e-default-rtdb.firebaseio.com/posts.",
        json", this.blog])
      .then(function (data) {
        console.log(data);
        this.submmited = true;
      });
  },
```

3. 更改showBlog.vue的get地址

```

created() {
  // 请求数据
  this.$http
    .get("https://blog-ad61e-default-rtdb.firebaseio.com/posts.json")
    .then(function (data) {
      // console.log(data);
      // 展示10个数据
      this.blogs = data.body.slice(0, 10);
      // console.log(this.blogs);
    });
}

```

#### 4. 将唯一标识转为id

```

created() {
  // 请求数据
  this.$http
    .get("https://blog-ad61e-default-rtdb.firebaseio.com/posts.json")
    .then(function (data) {
      // console.log(data.json());
      return data.json();
      // 展示10个数据
      // this.blogs = data.body.slice(0, 10);
      // console.log(this.blogs);
    })
    .then(function (data) {
      var blogsArray = [];
      // 得到唯一标识
      for (let key in data) {
        // console.log(key);
        // 得到每一个对象
        // console.log(data[key]);
        data[key].id = key;
        blogsArray.push(data[key]);
      }
      // console.log(blogsArray);
      this.blogs = blogsArray;
    });
}

```

将唯一标识转换为id

#### 5. 更改singleBlog.vue的地址

```

created() {
  this.$http
    .get(
      "https://blog-ad61e-default-rtdb.firebaseio.com/posts/" +
      this.id +
      ".json"
    )
}

```

#### 6. 处理数据

```

created() {
  this.$http
    .get(
      "https://blog-ad61e-default-rtdb.firebaseio.com/posts/" +
        this.id +
        ".json"
    )
    .then(function (data) {
      console.log(data);
      return data.json();
    })
    .then(function (data) {
      this.blog = data;
    });
},

```

**处理数据**

## 7. 展示数据

```

<template>
  <div id="single-blog">
    <h1>{{ blog.title }}</h1>
    <article>{{ blog.content }}</article>
    <p>作者: {{ blog.author }}</p>
    <p>分类: </p>
    <ul>
      <li v-for="category in blog.categories">
        {{ category }}
      </li>
    </ul>
  </div>
</template>

```

# 博客的编辑和删除功能

## 1. 绑定删除事件

```

<ul>
  <li v-for="category in blog.categories">
    {{ category }}
  </li>
</ul>
<button @click="deleteSingleBlog()"></button>
</div>

```

**绑定事件**

## 2. 设定删除方法

```

methods: {
  deleteSingleBlog() {
    this.$http.delete(
      "https://blog-ad61e-default-rtdb.firebaseio.com/posts/" + this.id + ".json"
    ).then(response=>{
      this.$router.push({path: '/'})
    })
  },
},

```

## 3. 设置编辑

```
<button @click="deleteSingleBlog()">删除</button>
<router-link :to="'/edit/' + id"></router-link>
</div>
</template>
```

设计编辑

#### 4. 设置路由

```
// 2.引入对象
import ShowBlogs from './components/ShowBlog.vue'
import AddBlog from './components/AddBlog.vue'
import SingleBlog from './components/SingleBlog.vue'
import EditBlog from './components/EditBlog.vue'
```

```
// 1.返回对象
```

```
export default [
```

```
{
  path: '/',
  component: ShowBlogs
},
```

```
{
  path: '/add',
  component: AddBlog
},
```

```
{
  path: '/blog/:id',
  component: SingleBlog
},
```

```
{
  path: "/edit/:id", component: EditBlog
}]
```

设置路由并引入

#### 5. editBlog.vue和addBlog.vue内容差多

```
1 <template>
2   <div id="add-blog">
3     <h2>编辑博客</h2>
4     <form v-if="!submmited">
5       <label>博客标题</label>
6       <input type="text" v-model="blog.title" required />
7       <label>博客内容</label>
8       <textarea v-model="blog.content"></textarea>
9
10      <div id="checkboxes">
11        <label>vue.js</label>
12        <input type="checkbox" value="vue.js" v-model="blog.categories"
13      />
14        <label>node.js</label>
15        <input type="checkbox" value="node.js" v-model="blog.categories"
16      />
17        <label>react.js</label>
18        <input type="checkbox" value="react.js" v-
19      model="blog.categories" />
20        <label>angular</label>
21        <input type="checkbox" value="angular" v-model="blog.categories"
22      />
23      </div>
24      <label>作者: </label>
25      <select v-model="blog.author">
26        <option v-for="author in authors">{{ author }}</option>
27      </select>
28      <button v-on:click.prevent="post">编辑博客</button>
29    </form>
30    <div v-if="submmited">
31      <h3>您的博客发布成功! </h3>
32    </div>
33    <div id="preview">
34      <h3>博客总览</h3>
35      <p>博客标题: {{ blog.title }}</p>
36      <p>博客内容: </p>
37      <p>{{ blog.content }}</p>
38      <p>博客分类: </p>
39      <ul>
40        <li v-for="category in blog.categories">{{ category }}</li>
41      </ul>
42      <p>作者: {{ blog.author }}</p>
43    </div>
44  </div>
45</template>
```



```

42
43 ▾ <script>
44 ▾ export default {
45     name: "add-blog",
46 ▾     data() {
47 ▾         return {
48 ▾             blog: {
49                 title: "",
50                 content: "",
51                 categories: [],
52                 author: "",
53             },
54             authors: ["小明", "小红", "小张", "小李"],
55             submmited: false,
56         };
57     },
58 ▾     methods: {
59 ▾         post: function () {
60             this.$http
61                 .post(
62                     "https://blog-ad61e-default-rtdb.firebaseio.com/posts.json",
63                     this.blog
64                 )
65 ▾                 .then(function (data) {
66                     console.log(data);
67                     this.submmited = true;
68                 });
69         },
70     },
71 };
72 </script>
73
74 <!-- Add "scoped" attribute to limit CSS to this component only -->
75 ▾ <style scoped>
76 ▾ #add-blog * {
77     box-sizing: border-box;
78 }
79 ▾ #add-blog {
80     margin: 20px auto;
81     max-width: 600px;
82     padding: 20px;
83 }
84
85 ▾ label {
86     display: block;
87     margin: 20px 0 10px;
88 }
89 input[type="text"],

```

```

90   textarea,
91   select {
92     display: block;
93     width: 100%;
94     padding: 8px;
95   }
96   textarea {
97     height: 200px;
98   }
99   #checkboxes label {
100     display: inline-block;
101     margin-top: 0;
102   }

```

## 6. 设置编辑页面中加载博客数据

```

<script>
export default {
  name: "add-blog",
  data() {
    return {
      id: this.$route.params.id,
      blog: {
        title: "",
        content: "",
        categories: [],
        author: "",
      },
      authors: ["小明", "小红", "小张", "小李"],
      submitted: false,
    };
  },
  methods: {
    fetchData() {
      console.log(this.id);
    },
    created() {
      // 请求数据
      this.fetchData();
    },
  },
}

```

设置方法请求数据后得到id

## 7. 拿取数据

```

methods: {
  fetchData() {
    // console.log(this.id);
    // 拿取数据
    this.$http
      .get(
        "https://blog-ad61e-default-rtdb.firebaseio.com/posts/" +
        this.id +
        ".json"
      )
      .then((response) => {
        // console.log(response);
      });
  },
}

```

拿取博客数据

## 8. 赋值

```
name: 'add-blog',
data() {
  return {
    id: this.$route.params.id,
    blog: {},
    authors: ["小明", "小红", "小张", "小李"],
    submitted: false,
  },
},
methods: {
  fetchData() {
    // console.log(this.id);
    // 拿取数据
    this.$http
      .get(
        "https://blog-ad61e-default-rtdb.firebaseio.com/posts/" +
        this.id +
        ".json"
      )
      .then((response) => {
        // console.log(response);
        this.blog = response.body;
      });
  }
},
}
```

设置blog为空后，把response里面的对象给它

## 9. 编辑完成put到数据库

```
post: function () {
  this.$http
    .put(
      "https://blog-ad61e-default-rtdb.firebaseio.com/posts/" +
      this.id +
      ".json",
      this.blog
    )
    .then(function (data) {
      console.log(data);
      this.submitted = true;
    });
},
}
```

编辑完成，put到数据库

# axios替换vue-resource

## 1. 安装axios

```
npm install axios
```

## 2. addBlog.vue中引入axios

```
import axios from "axios";
export default {
  name: "add-blog",
  data() {
    return {
      blog: {
        title: "",
        content: "",
        categories: [],
        author: "",
      },
    };
  },
}
```

## 3. 更改内容

```

methods: {
  post: function () {
    var _this = this;
    axios
      .post(
        "https://blog-ad61e-default-rtdb.firebaseio.com/posts.json",
        this.blog
      )
      .then(function (data) {
        console.log(data);
        _this.submmited = true;
      });
  },
},
};

```

es6箭头函数：

```

methods: {
  post: function () {
    axios
      .post(
        "https://blog-ad61e-default-rtdb.firebaseio.com/posts.json",
        this.blog
      )
      .then((data) => {
        console.log(data);
        this.submmited = true;
      });
  },
},
};

```

#### 4. 更改showBlogs.vue

```

import axios from "axios";

export default {
  name: "show-blogs",
  data() {
    return {
      blogs: [],
      search: "",
    };
  },
  created() {
    // 请求数据
    axios
      .get("https://blog-ad61e-default-rtdb.firebaseio.com/posts.json")
      .then(function (data) {
        // console.log(data.json());
        // return data.json();
        return data.data;
      })
      .then((data) => {
        var blogsArray = [];
        // 得到唯一标识
        for (let key in data) {
          // 得到每一个对象
          data[key].id = key;
          blogsArray.push(data[key]);
        }
        // console.log(blogsArray);
        this.blogs = blogsArray;
        console.log(this.blogs);
      });
  },
};

```

## 5. 更改SingleBlog.vue

```

import axios from "axios";

created() {
  axios
    .get(
      "https://blog-ad61e-default-rtdb.firebaseio.com/posts/" +
      this.id +
      ".json"
    )
    .then((data) => {
      // console.log(data);
      // return data.json();
      // return data.data;
      this.blog = data.data;
    });
};

```

```

methods: {
  deleteSingleBlog() {
    axios
      .delete(
        "https://blog-ad61e-default-rtdb.firebaseio.com/posts/" +
        this.id +
        ".json"
      )
      .then((response) => {
        this.$router.push({ path: "/" });
      });
  },
},
};

```

## 6. 全局配置axios

```

import App from './App'
import Routes from './routes'
import axios from 'axios'

// 全局配置
axios.defaults.baseURL = 'https://blog-ad61e-default-rtdb.firebaseio.com';

Vue.config.productionTip = false
// Vue.use(VueResource)
Vue.use(VueRouter)

```

## 7. 其他文件中删除网址

## 8. 配置请求头

```

import Routes from './routes'
import axios from 'axios'

// 全局配置
axios.defaults.baseURL = 'https://blog-ad61e-default-rtdb.firebaseio.com';
// 配置请求头
// axios.defaults.headers.common['Authorization'] = 'Token'
// axios.defaults.headers.post['Content-type'] = 'application/urlencode'
// axios.defaults.headers.get['Accepts'] = 'application/json'

```

# 配置axios

```
axios-auth.js > default
1 import axios from "axios";
2 const instance = axios.create({
3   baseURL: 'https://blog-ad61e-default-rtdb.firebaseio.com'
4 })
5
6
7 instance.defaults.headers.common['SOMETHING'] = 'SOMETHING'
8
9 export default instance
```

```
<script>
import axios from "../axios-auth";
```