勘误

res.sendStatus(401);

复习

- 路由守卫
- 登录状态保存
- 接口mock
- 请求、响应拦截器
- 令牌原理
    - 头.载荷.哈希
- 跨域和请求代理

内容

- 准备工作

  迁出second-seed分支：https://github.com/57code/vue-mart/tree/second-seed

  提供素材、mock接口、Cart、GoodList等

- 导航App.vue

```
 <cube-tab-bar v-model="selectLabel" :data="tabs"
@change="changeHandler">
      <cube-tab v-for="(item, index) in tabs"
            :icon="item.icon" :label="item.value" :key="index">
            <div>{{item.label}}</div>
            <span class="badge" v-if="item.label=='Cart'">
{{cartTotal}}</span>
        </cube-tab>
    </cube-tab-bar>


data() {
    return {
      transitionName: 'route-forward',
      selectLabel: "/",
      tabs: [
        {
          label: "Home",
          value: "/",
          icon: "cubeic-home"
        },
        {
          label: "Cart",
          value: "/cart",
          icon: "cubeic-mall"
        },
        {
          label: "Me",
          value: "/about",
          icon: "cubeic-person"
        }
      ]
    };
  },
  created() {
    // 初始化页签设置，避免页面刷新
    this.selectLabel = this.$route.path;
  },
  watch: {
    $route(route) {
      // 监听路由变化并动态设置页签选中状态
      this.selectLabel = route.path;

      this.transitionName = this.$router.transitionName
    }
  },
  methods: {
    changeHandler(val) {
      this.$router.push(val);
    }
  },
```

- 轮播图和商品列表
  - goods服务，service/goods.js

```
import axios from "axios";

export default {
  // 获取轮播图、商品和分类数据
  getGoodsInfo() {
    return axios.get("/api/goods").then(res => {
      const {code, data: goodsInfo, slider, keys} = res.data;
      // 数据处理
      if (code) {
        return { goodsInfo, slider, keys };
      } else {
        return null;
      }
    });
  }
};
```

- 轮播图、商品列表，Home.vue

```
<template>
    <!-- 轮播图 -->
    <cube-slide :data="slider" :interval="5000">
      <cube-slide-item v-for="(item,index) in slider" :key="index">
        <router-link :to="`/detail/${item.id}`">
          <img class="slider" :src="item.img">
        </router-link>
      </cube-slide-item>
    </cube-slide>

    <!-- 商品列表 -->
    <good-list :data="goods"></good-list>
</template>

<script>
import gs from "@/service/goods";
import GoodList from "@/components/GoodList.vue";

export default {
  name: "home",
  components: {
    GoodList
  },
  data() {
    return {
      slider: [],
      keys: [],
      goodsInfo: {}
    };
  },
  created() {
    // 首次查询
    gs.getGoodsInfo().then(({ goodsInfo, slider, keys }) => {
      // 保存状态
      this.goodsInfo = goodsInfo;
      this.slider = slider;
      this.keys = keys;
    });
  },
  computed: {
    goods() {
      return this.keys.flatMap(key => this.goodsInfo[key]);
    }
  }
};
</script>
```

购物车

- 购物车状态，cart.js

```
export default {
    state: { // 购物车初始状态
      list: JSON.parse(localStorage.getItem("cart")) || []
    },
    mutations: {
      addcart(state, item) { // 添加商品至购物车
        const good = state.list.find(v => v.title == item.title);
        if (good) {
          good.cartCount += 1;
        } else {
          state.list.push({
            ...item,
            cartCount: 1
          });
        }
      },
      cartremove(state, index) { // count-1
        if (state.list[index].cartCount > 1) {
          state.list[index].cartCount -= 1;
        }
      },
      cartadd(state, index) { // count+1
        state.list[index].cartCount += 1;
      }
    },
    getters: {
      cartTotal: state => { // 商品总数
        let num = 0;
        state.list.forEach(v => {
          num += v.cartCount;
        });
        return num;
      },
      total: state => { // 总价
```

```
          return state.list.reduce(
            (total, item) => total + item.cartCount * item.price,
            0
          );
        }
      }
  }
```

动画设计

- 添加购物车动画，CartAnim.vue

```
<template>
  <div class="ball-wrap">
    <transition @before-enter="beforeEnter" @enter="enter" @afterEnter="afterEnter">
      <div class="ball" v-show="show">
        <div class="inner">
          <div class="cubeic-add"></div>
        </div>
      </div>
    </transition>
  </div>
</template>
<script>
export default {
  name: "cartAnim",
  data() {
    return { show: false };
  },
  methods: {
    start(el) {// 启动动画接口，传递点击按钮元素
      this.el = el;

      // 使.ball显示，激活动画钩子
      this.show = true;
    },
    beforeEnter(el) {
      // 把小球移动到点击的dom元素所在位置
      const rect = this.el.getBoundingClientRect();
      // 转换为用于绝对定位的坐标
      const x = rect.left - window.innerWidth / 2;
      const y = -(window.innerHeight - rect.top - 10 - 20);

      // ball只移动y
      el.style.transform = `translate3d(0, ${y}px, 0)`;
      // inner只移动x
      const inner = el.querySelector(".inner");
      inner.style.transform = `translate3d(${x}px,0,0)`;
    },
    enter(el, done) {
      // 获取offsetHeight就会重绘
      document.body.offsetHeight;

      // 指定动画结束位置
      el.style.transform = `translate3d(0, 0, 0)`;
      const inner = el.querySelector(".inner");
      inner.style.transform = `translate3d(0,0,0)`;
      el.addEventListener("transitionend", done);
    },
    afterEnter(el) {
      // 动画结束，开始清理工作
      this.show = false;
      el.style.display = "none";
      this.$emit("transitionend");
    }
  }
};
</script>
<style scoped lang="stylus">
.ball-wrap {
  .ball {
    position: fixed;
    left: 50%;
    bottom: 10px;
    z-index: 100000;
    color: red;
    transition: all 0.5s cubic-bezier(0.49, -0.29, 0.75, 0.41);

    .inner {
      width: 16px;
      height: 16px;
      transition: all 0.5s linear;

      .cubeic-add {
        font-size: 22px;
      }
    }
  }
}
</style>
```

- 使用动画，Home.vue

```
<good-list @cartanim="$refs.ca.start($event)"></good-list>
<cart-anim ref="ca"></cart-anim>

import CartAnim from "@/components/CartAnim.vue";
components: { CartAnim }
```

- 触发动画，GoodList.vue

```
<i class="cubeic-add" @click.stop.prevent="addCart($event,item)"></i>

addCart(e,item) { // 需要传递事件目标
    this.$store.commit("addcart", item);
    // 触发动画事件
    this.$emit('cartanim',e.target)
}
```

动态全局组件设计与实现

- 使用cube-ui的create-api
  - 注册，main.js

```
import {createAPI} from 'cube-ui'
import CartAnim from '@/components/CartAnim'

createAPI(Vue, CartAnim, ['transitionend'])
```

  - 调用api，Home.vue

```
  <good-list :data="goods" @startcartanim="startCartAnim"></good-list>

  methods: {
    startCartAnim(el) {
      const anim = this.$createCartAnim({
        onTransitionend() {
          anim.remove();
        }
      });
      anim.start(el);
    }
  }
```

- 组件动态创建并挂载的具体实现./utils/create.js

```
import Vue from "vue";

// 创建函数接收要创建组件定义
function create(Component, props) {
  // 创建一个Vue新实例
  const instance = new Vue({
    render(h) {
      // render函数将传入组件配置对象转换为虚拟dom
      console.log(h(Component, { props }));
      return h(Component, { props });
    }
  }).$mount(); //执行挂载函数，但未指定挂载目标，表示只执行初始化、编译等工作
  // 将生成dom元素追加至body
  document.body.appendChild(instance.$el);
  // 给组件实例添加销毁方法
  const comp = instance.$children[0];
  comp.remove = () => {
    document.body.removeChild(instance.$el);
    instance.$destroy();
  };
  return comp;
}

// 暴露调用接口
export default create;
```

  - 挂载到vue实例上,main.js

```
import create from '@/utils/create'
Vue.prototype.$create = create;
```

  - 调用，Home.vue

```
startCartAnim(el) {
      const anim = this.$create(CartAnim);
      anim.start(el);
      anim.$on("transitionend", anim.remove);
}
```

页头组件

- 组件定义，Header.vue

```
<template>
  <div class="header">
    <h1>{{title}}</h1>
    <i v-if="showback" @click="back" class="cubeic-back"></i>
    <div class="extend">
      <slot></slot>
    </div>
  </div>
</template>

<script>
export default {
  props: {
    title: {
      type: String,
      default: "",
      required: true
    },
    showback: {
      type: Boolean,
      default: false
    }
  },
  methods: {
    back() {
      this.$router.back();
    }
  }
};
</script>
<style lang="stylus" scoped>
.header {
  position: relative;
  height: 44px;
  line-height: 44px;
  text-align: center;
  background: #edf0f4;

  .cubeic-back {
    position: absolute;
    top: 0;
    left: 0;
    padding: 0 15px;
    color: #fc915b;
  }

  .extend {
    position: absolute;
    top: 0;
    right: 0;
    padding: 0 15px;
    color: #fc915b;
  }
}
</style>
```

- 使用，Home.vue

```
<k-header title="开课吧">
    <i class="cubeic-tag"></i>
</k-header>

import KHeader from '@/components/Header.vue';
components: {
    GoodList,KHeader,
}
```

- 返回按钮状态自动判断：history.length是不可靠的，它既包含了vue app路由记录，也包括其他页面的。可以添加一个自定义的历史记录管理栈，创建./utils/history.js

```
const History = {
  _history: [], // 历史记录堆栈
  install(Vue) {
    // vue插件要求的安装方法
    Object.defineProperty(Vue.prototype, "$routerHistory", {
      get() {
        return History;
      }
    });
  },
  push(path) {
    // 入栈
    this._current += 1;
    this._history.push(path);
  },
  pop() {
    // 出栈
    this._current -= 1;
    return this._history.pop();
  },
  canBack() {
    return this._history.length > 1;
  }
};
export default History;
```

- router.js中引入，添加一个后退方法并监听afterEach从而管理记录

```
import History from "./utils/history";

Vue.use(History);

Router.prototype.goBack = function() {
  this.isBack = true;
  this.back();
};

router.afterEach((to, from) => {
  if (router.isBack) {
    History.pop();
    router.isBack = false;
    router.transitionName = "route-back";
  } else {
    History.push(to.path);
    router.transitionName = "route-forward";
  }
});
```

- 使用，Header.vue

```
<i v-if="$routerHistory.canBack()"></i>

methods: {
    back() {this.$router.goBack();}
}
```

- 后退动画，App.vue

```
<transition :name="transitionName">
    <router-view class="child-view"/>
</transition>

data() {
    return {
      transitionName: 'route-forward'
    };
  },
  watch: {
    $route() {
      // 动态设置动画方式
      this.transitionName = this.$router.transitionName
    }
  },


/* 页面滑动动画 */
/* 入场前 */
.route-forward-enter {
  transform: translate3d(-100%, 0, 0);
}
.route-back-enter {
  transform: translate3d(100%, 0, 0);
}
/* 出场后 */
.route-forward-leave-to {
  transform: translate3d(100%, 0, 0);
}
.route-back-leave-to {
  transform: translate3d(-100%, 0, 0);
}
.route-forward-enter-active,
.route-forward-leave-active,
.route-back-enter-active,
.route-back-leave-active {
  transition: transform 0.3s;
}
```