Vue 实战 -- 前端

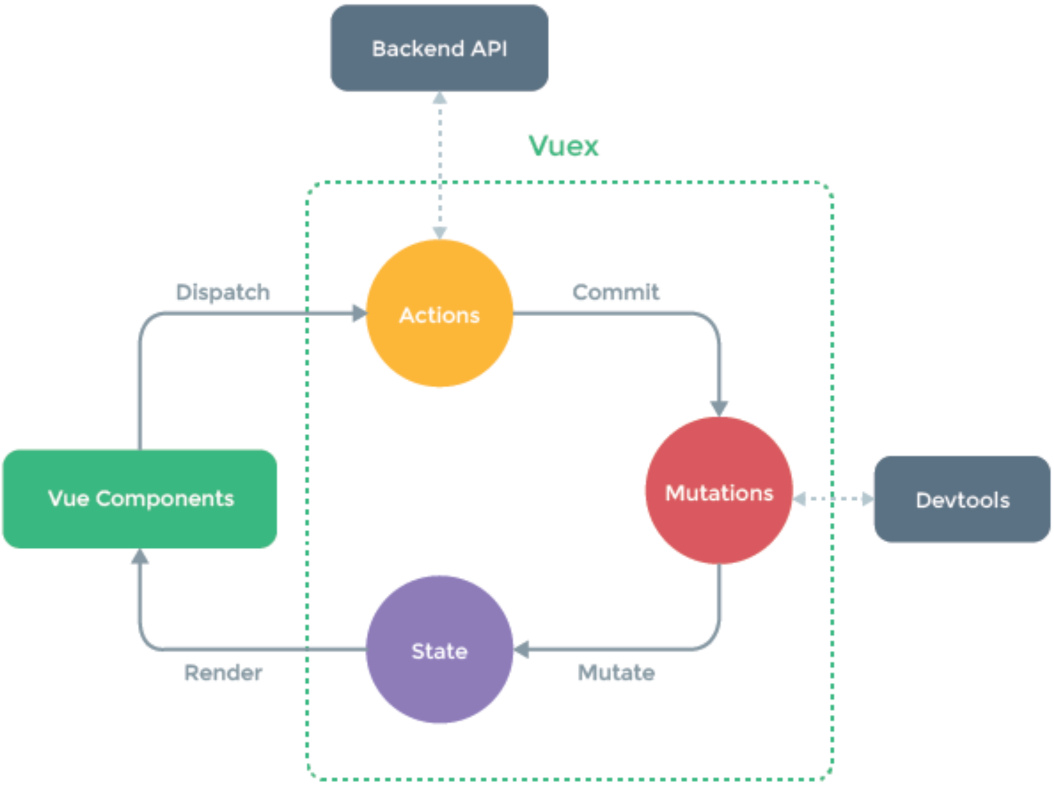
Vuex基本原理

Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式。它采用**集中式存储管理**应用的所有组件的状态，

这个状态管理应用包含以下几个部分：

- State，驱动应用的数据源；
- View，以声明方式将 state 映射到视图；
- Actions，响应在 view 上的用户输入导致的状态变化。

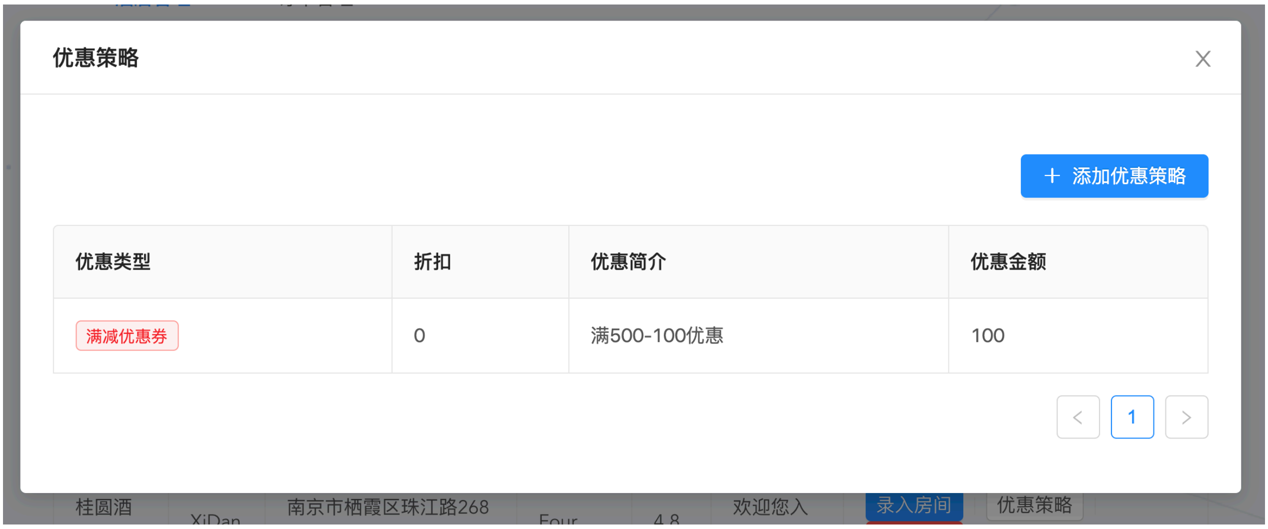
强调数据流的单向流动。



1. Vue页面**Commponent** -- 调用 *method* -- **State**修改-- render重新渲染页面**Commponent**
   - coupon页面添加优惠策略按钮
2. Vue页面**Commponent** -- 异步事件调用 *dispatch* --> **Action** --> 回调（可能包含dispatch，commit）
   - addcoupon页面确定按钮
3. Vue页面**Commponent** -- 同步调用*commit*一个type类型 --> **Mutation** -- 匹配对应type操作**State**-- *render*重新渲染页面**Commponent**
   - set_addHotelModalVisible设置页面的显示

List页面

coupon.vue中



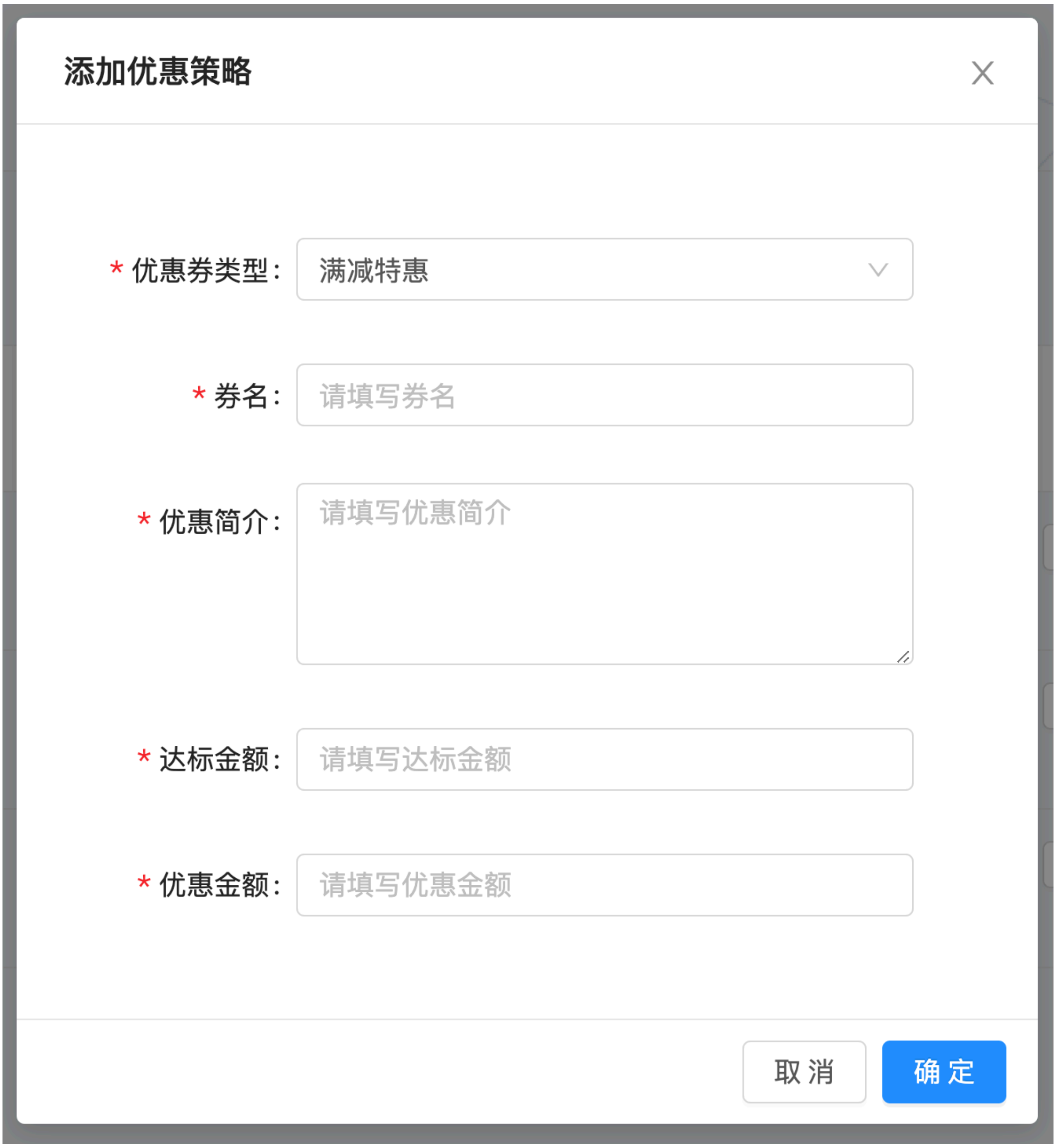
优惠策略

优惠类型	折扣	优惠简介	优惠金额
满减优惠券	0	满500-100优惠	100

```
<template>
   <div>
    <a-modal
        :visible="couponVisible"
        title="优惠策略"
        width="900px"
        :footer="null"
        @cancel="cancel"
    >
        <div style="width: 100%; text-align: right; margin:20px 0">
            <a-button type="primary" @click="addCoupon"><a-icon type="plus" />
添加优惠策略</a-button>  //click操作绑定addCoupon方法
        </div>
        <a-table
            :columns="columns"
            :dataSource="couponList"
            bordered
        >
            <a-tag color="red" slot="couponName" slot-scope="text">
                {{text}}
            </a-tag>
        </a-table>
    </a-modal>
    <AddCoupon></AddCoupon>
   </div>
</template>
<script>
import { mapGetters, mapMutations, mapActions } from 'vuex'
import AddCoupon from './addCoupon'

const columns = [
    {
        title: '优惠类型',
        dataIndex: 'couponName',
        scopedSlots: {customRender: 'couponName'}
    },
    {
        title: '折扣',
        dataIndex: 'discount',
    },
    {
        title: '优惠简介',
        dataIndex: 'description',

    },
    {
        title: '优惠金额',
        dataIndex: 'discountMoney',
    },
  ];
export default {
    name: 'coupon',
    data() {
        return {
            columns
        }
    },
    components: {
        AddCoupon,
    },
    computed: {
        ...mapGetters([
            'couponVisible',
            'couponList',
        ])
    },
    methods: {
        ...mapMutations([
            'set_addCouponVisible',
            'set_couponVisible',
        ]),
        ...mapActions([
            'getHotelCoupon'
        ]),
        cancel() {
            this.set_couponVisible(false)
        },
        addCoupon() {// 改变状态，自动渲染
            this.set_addCouponVisible(true),
            this.set_couponVisible(false)
        },
    },
}
</script>
<style scoped>

</style>
```

表单页面

addCoupon.vue



添加优惠策略

* 优惠券类型：满减特惠

* 券名：请填写券名

* 优惠简介：请填写优惠简介

* 达标金额：请填写达标金额

* 优惠金额：请填写优惠金额

```
<template>
    <a-modal
        :visible="addCouponVisible"
        title="添加优惠策略"
        cancelText="取消"
        okText="确定"
        @cancel="cancel"
        @ok="handleSubmit"
    >   <!-- #点确定的响应方法handleSubmit -->
        <a-form :form="form" style="margin-top: 30px" v-bind="formItemLayout">
            <a-form-item label="优惠券类型" v-bind="formItemLayout">
                <a-select
                    v-decorator="[
                    'type',
                    { rules: [{ required: true, message: '请选择类型' }] }]"
                    @change="changeType"
                >
                  <a-select-option value="1">生日特惠</a-select-option>
                  <a-select-option value="2">多间特惠</a-select-option>
                  <a-select-option value="3">满减特惠</a-select-option>
                  <a-select-option value="4">限时特惠</a-select-option>
                </a-select>
            </a-form-item>
            <a-form-item label="券名" v-bind="formItemLayout">
                <a-input
                    placeholder="请填写券名"
                    v-decorator="['name', { rules: [{ required: true, message: '请输入券名' }] }]"
                />
            </a-form-item>
            <a-form-item label="优惠简介" v-bind="formItemLayout">
                <a-input
                    type="textarea"
                    :rows="4"
                    placeholder="请填写优惠简介"
                    v-decorator="['description', { rules: [{ required: true, message: '请填写优惠简介' }] }]"
                />
            </a-form-item>
             <a-form-item label="达标金额">
                <a-input
                    placeholder="请填写达标金额"
                    v-decorator="['targetMoney', { rules: [{ required: true, message: '请填写达标金额' }] }]"
                />
            </a-form-item>
            <a-form-item label="优惠金额" v-bind="formItemLayout">
                <a-input
                    placeholder="请填写优惠金额"
                    v-decorator="['discountMoney', { rules: [{ required: true, message: '请填写优惠金额' }] }]"
                />
            </a-form-item>
        </a-form>
    </a-modal>
</template>
<script>
import { mapGetters, mapMutations, mapActions } from 'vuex'
export default {
    name: 'addCouponModal',
    data() {
        return {
            formItemLayout: {
                labelCol: {
                    xs: { span: 12 },
                    sm: { span: 6 },
                },
                wrapperCol: {
                    xs: { span: 24 },
                    sm: { span: 16 },
                },
            },
        }
    },
    computed: {
        ...mapGetters([
            'activeHotelId',
            'addCouponVisible',
        ])
    },
    beforeCreate() {
        // 表单名默认为“form”
        this.form = this.$form.createForm(this, { name: 'addCouponModal' });
    },
    mounted() {

    },
    methods: {
        ...mapMutations([
            // mutations  this.$store.commit(xxx) mapMutations
            'set_addCouponVisible'
        ]),
        ...mapActions([
            // addHotelCoupon：添加酒店优惠券接口
            //actions this.$store.dispatch（xxx ）  mapActions
            'addHotelCoupon'
        ]),
        cancel() {
            this.set_addCouponVisible(false)
        },
        changeType(v){
            if( v == '3') {

            }else{
                this.$message.warning('请实现')
            }
        },
        handleSubmit(e) {
            e.preventDefault();
            this.form.validateFieldsAndScroll((err, values) => {
                if (!err) {
                    const data = {
                        name: this.form.getFieldValue('name'),
                        description: this.form.getFieldValue('description'),
                        type: Number(this.form.getFieldValue('type')),
                        targetMoney:
Number(this.form.getFieldValue('targetMoney')),
                        discountMoney:
Number(this.form.getFieldValue('discountMoney')),
                        hotelId: Number(this.activeHotelId),
                        status: 1,
                    }
                    this.addHotelCoupon(data)
                }
            });
        },//响应按钮的按下，调用action方法
    }
}
</script>
```

js

store文件夹下hotelManager.js

```
import {
    addRoomAPI,
    addHotelAPI,
} from '@/api/hotelManager'
import {
    getAllOrdersAPI,
} from '@/api/order'
import {
    hotelAllCouponsAPI,
    hotelTargetMoneyAPI,      //导入api的定义
} from '@/api/coupon'
import { message } from 'ant-design-vue'

const hotelManager = {
    state: {
        orderList: [],
        addHotelParams: {
            name: '',
            address: '',
            bizRegion:'XiDan',
            hotelStar:'',
            rate: 0,
            description:'',
            phoneNum:'',
            managerId:'',
        },
        addHotelModalVisible: false,
        addRoomParams: {
            roomType: '',
            hotelId: '',
            price: '',
            total: 0,
            curNum: 0,
        },
        addRoomModalVisible: false,
        couponVisible: false,
        addCouponVisible: false,
        activeHotelId: 0,
        couponList: [],
    },
    mutations: {
        set_orderList: function(state, data) {
            state.orderList = data
        },
        set_addHotelModalVisible: function(state, data) {
            state.addHotelModalVisible = data
        },
        set_addHotelParams: function(state, data) {
            state.addHotelParams = {
                ...state.addHotelParams,
                ...data,
            }
        },
        set_addRoomModalVisible: function(state, data) {
            state.addRoomModalVisible = data
        },
        set_addRoomParams: function(state, data) {
            state.addRoomParams = {
                ...state.addRoomParams,
                ...data,
            }
        },
        set_couponVisible: function(state, data) {
            state.couponVisible = data
        },
        set_activeHotelId: function(state, data) {
            state.activeHotelId = data
        },
        set_couponList: function(state, data) {
            state.couponList = data
        },
        set_addCouponVisible: function(state, data) {
            state.addCouponVisible =data
        }
    },
    actions: {
        getAllOrders: async({ state, commit }) => {
            const res = await getAllOrdersAPI()
            if(res){
                commit('set_orderList', res)
            }
        },
        addHotel: async({ state, dispatch, commit }) => {
            const res = await addHotelAPI(state.addHotelParams)
            if(res){
                dispatch('getHotelList')
                commit('set_addHotelParams', {
                    name: '',
                    address: '',
                    bizRegion:'XiDan',
                    hotelStar:'',
                    rate: 0,
                    description:'',
                    phoneNum:'',
                    managerId:'',
                })
                commit('set_addHotelModalVisible', false)
                message.success('添加成功')
            }else{
                message.error('添加失败')
            }
        },
        addRoom: async({ state, dispatch, commit }) => {
            const res = await addRoomAPI(state.addRoomParams)
            if(res){
                commit('set_addRoomModalVisible', false)
                commit('set_addRoomParams', {
                    roomType: '',
                    hotelId: '',
                    price: '',
                    total: 0,
                    curNum: 0,
                })
                message.success('添加成功')
            }else{
                message.error('添加失败')
            }
        },
        getHotelCoupon: async({ state, commit }) => {
            const res = await hotelAllCouponsAPI(state.activeHotelId)
            if(res) {
                commit('set_couponList', res)
            }
        },
        addHotelCoupon: async({ commit, dispatch }, data) => {
            const res = await hotelTargetMoneyAPI(data)
            if(res){
                dispatch('getHotelCoupon')
                commit('set_addCouponVisible', false)
                commit('set_couponVisible',true)
                message.success('添加策略成功')
            }else{
                message.error('添加失败')
            }
        }//异步的方法调用，访问服务器api，以及回调方法（包括新的dispatch 去取酒店的所有coupon信息和commit view跳转）
    }
}
export default hotelManager
```

api文件夹下coupon.js文件

```
import { axios } from '@/utils/request'
const api = {
    couponPre: '/api/coupon'
}
export function hotelTargetMoneyAPI(data) {
    return axios({
        url: `${api.couponPre}/hotelTargetMoney`,
        method: 'POST',
        data,
    })//POST请求添加coupon
}

export function hotelAllCouponsAPI(hotelId) {
    return axios({
        url: `${api.couponPre}/hotelAllCoupons`,
        method: 'GET',
        params: {hotelId: hotelId},
    })
}

export function orderMatchCouponsAPI(params) {
    return axios({
        url: `${api.couponPre}/orderMatchCoupons`,
        method: 'GET',
        params,
    })
}
```

controller

```
package com.example.hotel.controller.coupon;

import com.example.hotel.bl.coupon.CouponService;
import com.example.hotel.vo.CouponVO;
import com.example.hotel.vo.HotelTargetMoneyCouponVO;
import com.example.hotel.vo.OrderVO;
import com.example.hotel.vo.ResponseVO;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/api/coupon")
public class CouponController {

    @Autowired
    private CouponService couponService;

    @PostMapping("/hotelTargetMoney")//添加coupon，调用逻辑层接口
    public ResponseVO addHotelTargetMoneyCoupon(@RequestBody
HotelTargetMoneyCouponVO hotelTargetMoneyCouponVO) {

        CouponVO couponVO =
couponService.addHotelTargetMoneyCoupon(hotelTargetMoneyCouponVO);

        return ResponseVO.buildSuccess(couponVO);
    }

    @GetMapping("/hotelAllCoupons")
    public ResponseVO getHotelAllCoupons(@RequestParam Integer hotelId) {
        return
ResponseVO.buildSuccess(couponService.getHotelAllCoupon(hotelId));
    }

    @GetMapping("/orderMatchCoupons")
    public ResponseVO getOrderMatchCoupons(@RequestParam Integer userId,
                                           @RequestParam Integer hotelId,
                                           @RequestParam Double orderPrice,
                                           @RequestParam Integer roomNum,
                                           @RequestParam String checkIn,
                                           @RequestParam String checkOut) {
        OrderVO requestOrderVO = new OrderVO();
        requestOrderVO.setUserId(userId);
        requestOrderVO.setHotelId(hotelId);
        requestOrderVO.setPrice(orderPrice);
        requestOrderVO.setRoomNum(roomNum);
        requestOrderVO.setCheckInDate(checkIn);
        requestOrderVO.setCheckOutDate(checkOut);
        return
ResponseVO.buildSuccess(couponService.getMatchOrderCoupon(requestOrderVO));
    }


}
```