**一.验证登录表单**

**1.给from元素设置rules属性，在给管理每一个输入框的父元素添加prop属性，值为需要效验的字段名，**

**rules 规则对象配置到 el-form 上，prop 校验字段配置到 el-form-item 上，校验规则写在data里面，用rules定义，值为对象，每一项的属性是需要效验的字段名，值为一个数组，数组里传入的是两个对象，一个是效验的内容为空时提示的信息，一个是设置效验内容的格式，如果不符合提示的内容**

**rules: {**

**mobile: [**

**{ required: true, message: '请输入手机号', trigger: 'blur' },**

**{ pattern: /^1([38]\d|5[0-35-9]|7[3678])\d{8}$/, message: '请输入正确手机号', trigger: 'blur' }**

**],，**

**2.在点击发送验证码时再次验证，验证表单的部分字段（此时验证的手机号内容跟离开焦点一样的验证规则，是因为鼠标离开焦点时只是提示，点击获取验证码还是会向后台发送请求，点击获取二维码时所验证里面的内容才是最主要的，如果输入的内容不符合效验，不会往下一步进行，阻止了ajax的请求发送，减少了服务器的请求），用表单的方法validateField（），里面是两个参数，第一个参数是需要验证的表单字段名，第二个参数是成功或者失败后返回的数据，成功无返回内容，失败有消息字符串返回，可以判断字数来确定是否成功，如果成功，在发送ajax到人机验证码端口，展示人机验证码，**

**3.其次，用户协议和自定义效验，给checkbox绑定一个v-model交互和prop属性字段名，加入效验，用正则验证，/true/，**

**4.给获取验证码添加倒计时，需要的内容有倒计时期间要禁用disabled，内容变化，倒计时结束禁用结束，内容再次改变， 定义一个计时器名称默认null和一个初始化时间，当点击发送验证码之后，定时器名变为数字，转换为blooen值为true，内容变换，把disabled的值转换为true，时间结束之后，把计时器名称赋值为null，时间再次初始化**

**，在组件中，this的指向为当前的组件，但是默认函数中的this指向的是window，所以要把默认函数改变成箭头函数，箭头函数中没有this，this指向上一级作用域**

**5.最后，点击登录按钮时，再次验证整体表单，如果验证成功，可以发送请求，验证不成功，禁止发送请求，跟验证验证码时一样，第一次只是提示，这次验证才会阻止，做阻止发送请求验证时会先获取子组件的dom，此时可以给表单组件添加ref属性，用this.$refs['ref字段名']获取值**

**二.创建并配置layout组件路由，给每个单组件页面都设置name值**

**给后台模块创建一个单独的文件夹，layout，后台模块中的样式很相似，边部跟头部一样，main部分不同，把跳转的界面设置为主页index.vue，边部头部和内容区域都作为子组件经过index.vue的渲染展示，其他的内容都经过layout中的index.vue展示其他各个后台页面中的main组件，这种其他后台的界面定义为子模块，**

**在后台的主页面中引入头部和边部的组件内容，给主页定义子组件，把引入的头部和边部传递进去，然后在主页的模板中加载，**

**三.用布局容器添加边部内容和导航菜单**

**在E-UI中查找匹配的布局容器，添加到主页index.vue模板中，查找合适的边栏和头部放到边栏和头部的单页面组件中**

**四.嵌套路由**

**在路由界面中，给一个路由添加子路由，属性名为children值为数组，数组中的每一个值都是对象，对象里面定义了子路由的名称，访问url路径和模板地址，子路由是以父路由为模板，此时需要在父路由中在定义一个路由入口，让子路由的内容在父路由的哪个区域展示，如果子路由没写url路径，会在访问父路由时自动渲染在模板中，创建文章板块publish，文章的主页为发表文章，边部组件中每一个小按钮都有一个唯一的index值，index 用来区分不同的导航项目的，可以用作路由导航路径，必须唯一，否则会出现菜单冲突**，**可以在边部组件的最高级中添加:router="true"属性，添加了这个属性后可以根据index的值跳转到相对应的页面，前提是这个页面在路由表中已经添加，**

**五.登录时显示用户信息**

**在login/index.vue页面中，登录成功之后，跳转页面会跳转到后台的主页面，登录成功之后又返回值，里面有用户登录的信息内容，可以把此时获取到的用户信息存储到本地存储中，本地存储就是把数据存储到浏览器中，和服务器端没有任何关系，只是浏览器的一种能力，只能存储字符串，如果不是会自动转换为字符串，如果要存储对象或者数据，可以把对象转换位json字符串，取出的时候在把json字符串转换为对象，**

**在此项目中，在登录成功界面window.localStorage.setItem(保存之后的数据名称，JSON.stringify(对象))**const userInfo = res.data.data

window.localStorage.setItem('user\_info', JSON.stringify(userInfo))

然后在头部组件中的初始化完成阶段拿出这个内容，created () {

this.userInfo = JSON.parse(window.localStorage.getItem('user\_info'))

}，**赋值到头部框内**