**node\_modules\vite-plugin-babel-import\example\src\App.vue**

<template>
  
 <img alt="Vue logo" src="./assets/logo.png" />
  
 <HelloWorld msg="Hello Vue 3 + Vite" />
  
 <div class="flex">
  
 <div>
  
 <van-button type="default">default</van-button>
  
 <van-button type="primary">primary</van-button>
  
 <van-button type="success">success</van-button>
  
 <van-button type="info">info</van-button>
  
 <van-button type="warning">warning</van-button>
  
 <van-button type="danger">danger</van-button>
  
 </div>
  
  
 <div>
  
 <el-button>default</el-button>
  
 <el-button type="primary">primary</el-button>
  
 <el-button type="success">success</el-button>
  
 <el-button type="info">info</el-button>
  
 <el-button type="warning">warning</el-button>
  
 <el-button type="danger">danger</el-button>
  
 </div>
  
  
 <div>
  
 <a-tag color="pink"> pink </a-tag>
  
 <a-tag color="red"> red </a-tag>
  
 <a-tag color="orange"> orange </a-tag>
  
 <a-tag color="green"> green </a-tag>
  
 <a-tag color="cyan"> cyan </a-tag>
  
 <a-tag color="blue"> blue </a-tag>
  
 <a-tag color="purple"> purple </a-tag>
  
 <a-badge count="5">
  
 <a href="#" class="head-example" />
  
 </a-badge>
  
 </div>
  
 </div>
  
</template>
  
  
<script>
  
import { Button } from 'vant';
  
import { ElButton } from 'element-plus';
  
import { Tag } from 'ant-design-vue';
  
  
import HelloWorld from './components/HelloWorld.vue';
  
  
export default {
  
 name: 'PageHome',
  
 components: {
  
 [Button.name]: Button,
  
 [ElButton.name]: ElButton,
  
 [Tag.name]: Tag,
  
 HelloWorld,
  
 },
  
 setup() {
  
 return {};
  
 },
  
};
  
</script>
  
  
<style>
  
#app {
  
 font-family: Avenir, Helvetica, Arial, sans-serif;
  
 -webkit-font-smoothing: antialiased;
  
 -moz-osx-font-smoothing: grayscale;
  
 text-align: center;
  
 color: #2c3e50;
  
 margin-top: 60px;
  
}
  
  
.flex {
  
 display: flex;
  
 flex-direction: column;
  
}
  
  
.flex > div {
  
 margin-top: 20px;
  
}
  
</style>

**node\_modules\vite-plugin-babel-import\example\src\components\HelloWorld.vue**

<template>
  
 <h1>{{ msg }}</h1>
  
  
 <p>
  
 <a href="https://vitejs.dev/guide/features.html" target="\_blank">Vite Documentation</a> |
  
 <a href="https://v3.vuejs.org/" target="\_blank">Vue 3 Documentation</a>
  
 </p>
  
  
 <button @click="state.count++">count is: {{ state.count }}</button>
  
 <p>
  
 Edit
  
 <code>components/HelloWorld.vue</code> to test hot module replacement.
  
 </p>
  
</template>
  
  
<script setup>
  
import { defineProps, reactive } from 'vue'
  
  
defineProps({
  
 msg: String
  
})
  
  
const state = reactive({ count: 0 })
  
</script>
  
  
<style scoped>
  
a {
  
 color: #42b983;
  
}
  
</style>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\App.vue**

<script setup lang="ts">  
import { useHead } from '@vueuse/head'  
  
// https://github.com/vueuse/head  
// you can use this to manipulate the document head in any components,  
// they will be renedered correctly in the html results with vite-ssg  
useHead({  
 title: 'Vitesse',  
 meta: [  
 { name: 'description', content: 'Opinionated Vite Starter Template' },  
 ],  
})  
</script>  
  
<template>  
 <ElButton type="danger">危险按钮</ElButton>  
 <router-view />  
</template>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\components\Footer.vue**

<script setup lang="ts">  
import { useI18n } from 'vue-i18n'  
import { isDark, toggleDark } from '~/logics'  
  
const { t, availableLocales, locale } = useI18n()  
  
const toggleLocales = () => {  
 // change to some real logic  
 const locales = availableLocales  
 locale.value = locales[(locales.indexOf(locale.value) + 1) % locales.length]  
}  
</script>  
  
<template>  
 <nav class="text-xl mt-6">  
 <router-link class="icon-btn mx-2" to="/" :title="t('button.home')">  
 <carbon-campsite />  
 </router-link>  
  
 <a class="icon-btn mx-2" :title="t('button.toggle\_dark')" @click="toggleDark">  
 <carbon-moon v-if="isDark" />  
 <carbon-sun v-else />  
 </a>  
  
 <a class="icon-btn mx-2" :title="t('button.toggle\_langs')" @click="toggleLocales">  
 <carbon-language />  
 </a>  
  
 <router-link class="icon-btn mx-2" to="/about" :title="t('button.about')">  
 <carbon-dicom-overlay />  
 </router-link>  
  
 <a class="icon-btn mx-2" rel="noreferrer" href="https://github.com/antfu/vitesse" target="\_blank" title="GitHub">  
 <carbon-logo-github />  
 </a>  
 </nav>  
</template>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\layouts\404.vue**

<script setup lang="ts">  
import { useRouter } from 'vue-router'  
import { useI18n } from 'vue-i18n'  
  
const router = useRouter()  
const { t } = useI18n()  
</script>  
  
<template>  
 <main class="px-4 py-10 text-center text-teal-700 dark:text-gray-200">  
 <div>  
 <p class="text-4xl">  
 <carbon-warning class="inline-block" />  
 </p>  
 </div>  
 <router-view />  
 <div>  
 <button  
 class="btn m-3 text-sm mt-8"  
 @click="router.back()"  
 >  
 {{ t('button.back') }}  
 </button>  
 </div>  
 </main>  
</template>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\layouts\default.vue**

<template>  
 <main class="px-4 py-10 text-center text-gray-700 dark:text-gray-200">  
 <router-view />  
 <Footer />  
 <div class="mt-5 mx-auto text-center opacity-25 text-sm">  
 [Default Layout]  
 </div>  
 </main>  
</template>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\layouts\home.vue**

<template>  
 <main class="px-4 py-10 text-center text-gray-700 dark:text-gray-200">  
 <router-view />  
 <Footer />  
 <div class="mt-5 mx-auto text-center opacity-25 text-sm">  
 [Home Layout]  
 </div>  
 </main>  
</template>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\pages\demo.vue**

<template>  
 <ElRow>  
 <ElButton>默认按钮</ElButton>  
 <ElButton type="primary">主要按钮</ElButton>  
 <ElButton type="success">成功按钮</ElButton>  
 <ElButton type="info">信息按钮</ElButton>  
 <ElButton type="warning">警告按钮</ElButton>  
 <ElButton type="danger">危险按钮</ElButton>  
 </ElRow>  
</template>  
  
<script>  
import { ElButton, ElRow } from 'element-plus'  
  
export default {  
 components: {  
 ElRow,  
 ElButton,  
 },  
}  
</script>  
  
<route lang="yaml">  
meta:  
 layout: home  
</route>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\pages\index.vue**

<script setup lang="ts">  
import { ref } from 'vue'  
import { useRouter } from 'vue-router'  
import { useI18n } from 'vue-i18n'  
  
const name = ref('')  
  
const router = useRouter()  
const go = () => {  
 if (name.value)  
 router.push(`/hi/${encodeURIComponent(name.value)}`)  
}  
  
const { t } = useI18n()  
</script>  
  
<template>  
 <div>  
 <p class="text-4xl">  
 <carbon-campsite class="inline-block" />  
 </p>  
 <p>  
 <a rel="noreferrer" href="https://github.com/antfu/vitesse" target="\_blank">  
 Vitesse  
 </a>  
 </p>  
 <p>  
 <em class="text-sm opacity-75">{{ t('intro.desc') }}</em>  
 </p>  
  
 <div class="py-4" />  
  
 <input  
 id="input"  
 v-model="name"  
 :placeholder="t('intro.whats-your-name')"  
 :aria-label="t('intro.whats-your-name')"  
 type="text"  
 autocomplete="false"  
 class="px-4 py-2 text-sm text-center bg-transparent border border-gray-200 rounded outline-none active:outline-none dark:border-gray-700"  
 style="width: 250px"  
 @keydown.enter="go"  
 >  
 <label class="hidden" for="input">{{ t('intro.whats-your-name') }}</label>  
  
 <div>  
 <button  
 class="m-3 text-sm btn"  
 :disabled="!name"  
 @click="go"  
 >  
 {{ t('button.go') }}  
 </button>  
 </div>  
 </div>  
</template>  
  
<route lang="yaml">  
meta:  
 layout: home  
</route>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\pages[...all].vue**

<script setup lang="ts">  
import { useI18n } from 'vue-i18n'  
  
const { t } = useI18n()  
</script>  
  
<template>  
 <div>  
 {{ t('not-found') }}  
 </div>  
</template>  
  
<route lang="yaml">  
meta:  
 layout: 404  
</route>

**node\_modules\vite-plugin-babel-import\example-vitesse\src\pages\hi[name].vue**

<script setup lang="ts">  
import { useRouter } from 'vue-router'  
import { useI18n } from 'vue-i18n'  
import { defineProps } from 'vue'  
  
const props = defineProps({  
 name: {  
 type: String,  
 required: true,  
 },  
})  
  
const router = useRouter()  
const { t } = useI18n()  
</script>  
  
<template>  
 <div>  
 <p class="text-4xl">  
 <carbon-pedestrian class="inline-block" />  
 </p>  
 <p>  
 {{ t('intro.hi', { name: props.name }) }}  
 </p>  
 <p class="text-sm opacity-50">  
 <em>{{ t('intro.dynamic-route') }}</em>  
 </p>  
  
 <div>  
 <button  
 class="btn m-3 text-sm mt-8"  
 @click="router.back()"  
 >  
 {{ t('button.back') }}  
 </button>  
 </div>  
 </div>  
</template>

**src\App.vue**

<script setup lang="ts">  
import { RouterLink, RouterView } from 'vue-router'  
import Header from './components/Header.vue'  
import Footer from './components/Footer.vue'  
import { reactive, toRefs } from 'vue'  
import { useRouter } from 'vue-router'  
  
// 不需要菜单的路径数组  
const noMenu = ['/login']  
const router = useRouter()  
const state = reactive({  
 showMenu: true, // 是否需要显示菜单  
})  
// 监听路由的变化  
router.beforeEach((to) => {  
 state.showMenu = !noMenu.includes(to.path)  
})  
</script>  
  
<template>  
 <div class="layout">  
 <el-container v-if="state.showMenu" class="container">  
 <el-aside class="aside">  
 <!--系统名称+logo-->  
 <div class="head">  
 <div>  
 <!-- <img src="./assets/logo.svg" alt="logo"> -->  
 <span>环境监控</span>  
 </div>  
 </div>  
 <!--一条为了美观的线条-->  
 <div class="line" />  
 <el-menu background-color="#222832" text-color="#fff" router="true">  
 <!--一级栏目-->  
 <el-sub-menu index="1">  
 <template #title>  
 <span>Dashboard</span>  
 </template>  
 <!--二级栏目-->  
 <el-menu-item-group>  
 <el-menu-item index="/"><el-icon>  
 <DataLine />  
 </el-icon>首页</el-menu-item>  
 <el-menu-item index="/login"><el-icon>  
 <DataLine />  
 </el-icon>登录</el-menu-item>  
 <el-menu-item index="/data"><el-icon>  
 <User />  
 </el-icon>数据管理</el-menu-item>  
 </el-menu-item-group>  
 </el-sub-menu>  
 </el-menu>  
 </el-aside>  
 <el-container class="content">  
 <Header />  
 <div class="main">  
 <!--将 <router-view></router-view> 移到这里，并且用单标签-->  
 <router-view />  
 </div>  
 <Footer />  
 </el-container>  
 </el-container>  
 <el-container v-else class="container">  
 <router-view />  
 </el-container>  
 </div>  
</template>  
  
<style scoped>  
.layout {  
 min-height: 100vh;  
 background-color: #ffffff;  
}  
  
.container {  
 height: 100vh;  
}  
  
.aside {  
 width: 200px !important;  
 background-color: #222832;  
}  
  
.head {  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 height: 50px;  
}  
  
.head>div {  
 display: flex;  
 align-items: center;  
}  
  
.head img {  
 width: 50px;  
 height: 50px;  
 margin-right: 10px;  
}  
  
.head span {  
 font-size: 20px;  
 color: #ffffff;  
}  
  
.line {  
 border-top: 1px solid hsla(0, 0%, 100%, .05);  
 border-bottom: 1px solid rgba(0, 0, 0, .2);  
}  
  
.content {  
 display: flex;  
 flex-direction: column;  
 max-height: 100vh;  
 overflow: hidden;  
}  
  
.main {  
 height: calc(100vh - 100px);  
 overflow: auto;  
 padding: 10px;  
}  
</style>  
  
<style>  
body {  
 padding: 0;  
 margin: 0;  
 box-sizing: border-box;  
}  
</style>

**src\components\Footer.vue**

<!--Footer.vue-->  
<template>  
 <div class="footer">  
 <div class="left">源码更新中...</div>  
 <div class="right">  
 <a target="\_blank" href="https://gitee.com/lemon-life/environmental-monitoring.git">environmental-monitoring</a>  
 </div>  
 </div>  
</template>  
  
<script setup lang="ts">  
// export default {  
// name: 'Footer'  
// }  
</script>  
  
<style scoped>  
 .footer {  
 height: 50px;  
 border-top: 1px solid #e9e9e9;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 0 20px;  
 }  
</style>

**src\components\Header.vue**

<!--Header.vue-->  
<template>  
 <div class="header">  
 <div class="left">  
 <span style="font-size: 20px">{{ state.name }}</span>  
 </div>  
 <div class="right">  
 <el-popover placement="bottom" :width="320" trigger="click" popper-class="popper-user-box">  
 <template #reference>  
 <div class="author">  
 <i class="icon el-icon-s-custom" />  
 {{ state.userInfo && state.userInfo.username|| '' }}  
 <i class="el-icon-caret-bottom" />  
 </div>  
 </template>  
 <div class="nickname">  
 <p>登录名：{{ state.userInfo && state.userInfo.username|| '' }}</p>  
 <!-- <p>昵称：{{ state.userInfo && state.userInfo.name|| '' }}</p> -->  
 <el-tag size="small" effect="dark" class="logout" @click="logout">退出</el-tag>  
 </div>  
 </el-popover>  
 </div>  
 </div>  
</template>  
  
<script setup lang="ts">  
// export default {  
// name: 'Header'  
// }  
import { reactive, toRefs } from 'vue'  
import { useRouter } from 'vue-router'  
// 获取路由实例  
const router = useRouter()  
// 声明路由和 title 对应的键值对  
const pathMap = {  
 index: '首页',  
 add: '添加数据记录'  
}  
const state = reactive({  
 name: '首页'  
})  
// 监听路由变化方法 afterEach  
router.afterEach((to) => {  
 console.log('to', to)  
 // to 能获取到路由相关信息。  
 const { id } = to.query  
 state.name = pathMap[to.name]  
 state.userInfo = JSON.parse(localStorage.getItem('userInfo') || '{}')  
 console.log('state.userInfo', state.userInfo)  
})  
</script>  
  
<style scoped>  
.header {  
 height: 50px;  
 border-bottom: 1px solid #e9e9e9;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 0 20px;  
}  
  
.right>div>.icon {  
 font-size: 18px;  
 margin-right: 6px;  
}  
  
.author {  
 margin-left: 10px;  
 cursor: pointer;  
}  
</style>  
  
<style>  
 .popper-user-box {  
 background: url('https://s.yezgea02.com/lingling-h5/static/account-banner-bg.png') 50% 50% no-repeat!important;  
 background-size: cover!important;  
 border-radius: 0!important;  
 }  
 .popper-user-box .nickname {  
 position: relative;  
 color: #ffffff;  
 }  
 .popper-user-box .nickname .logout {  
 position: absolute;  
 right: 0;  
 top: 0;  
 cursor: pointer;  
 }  
</style>

**src\components\HelloWorld.vue**

<script setup lang="ts">  
defineProps<{  
 msg: string  
}>()  
</script>  
  
<template>  
 <div class="greetings">  
 <h1 class="green">{{ msg }}</h1>  
 <h3>  
 You’ve successfully created a project with  
 <a href="https://vitejs.dev/" target="\_blank" rel="noopener">Vite</a> +  
 <a href="https://vuejs.org/" target="\_blank" rel="noopener">Vue 3</a>. What's next?  
 </h3>  
 </div>  
</template>  
  
<style scoped>  
h1 {  
 font-weight: 500;  
 font-size: 2.6rem;  
 top: -10px;  
}  
  
h3 {  
 font-size: 1.2rem;  
}  
  
.greetings h1,  
.greetings h3 {  
 text-align: center;  
}  
  
@media (min-width: 1024px) {  
 .greetings h1,  
 .greetings h3 {  
 text-align: left;  
 }  
}  
</style>

**src\components\Table.vue**

<template>  
 <el-table  
 :load="state.loading"  
 :data="state.tableData"  
 tooltip-effect="dark"  
 style="width: 100%"  
 @selection-change="handleSelectionChange"  
 >  
 <slot name='column'></slot>  
 </el-table>  
 <el-pagination  
 background  
 layout="prev, pager, next"  
 :total="state.total"  
 :page-size="state.pageSize"  
 :current-page="state.currentPage"  
 @current-change="changePage"  
 />  
</template>  
  
<script setup>  
import { onMounted, reactive } from 'vue'  
import axios from 'axios'  
  
const props = defineProps({  
 action: String  
})  
const state = reactive({  
 loading: false,  
 tableData: [], // 数据列表  
 total: 0, // 总条数  
 currentPage: 1, // 当前页  
 pageSize: 10, // 分页大小  
 multipleSelection: [], // 多选框  
 })  
 // 初始化钩子函数  
 onMounted(() => {  
 getList()  
 })  
  
 // 获取列表方法  
 const getList = () => {  
 state.loading = true  
 axios.get("/api/" + props.action, {  
 params: {  
 page: state.currentPage,  
 size: state.pageSize,  
 sort: ''  
 }  
 }).then(res => {  
 console.log(res)  
 state.tableData = res.data.content  
 console.log(res.data.content)  
 state.total = res.data.totalElements  
 console.log(res.data.totalElements)  
 // 由于组件绑定时是从1开始，但是后端需要从0开始，所以根据后端返回的信息需要加1  
 state.currentPage = res.data.number + 1  
 console.log(res.data.number)  
 state.loading = false  
 })  
 }  
 // 选项  
 const handleSelectionChange = (val) => {  
 state.multipleSelection = val  
 }  
 // 分页方法  
 const changePage = (val) => {  
 state.currentPage = val  
 getList()  
 }  
 // script setup 写法，需要通过 defineExpose 方法，将属性暴露出去，才能在父组件通过 ref 形式拿到本组件的内部参数  
 defineExpose({ state: state, getList: getList })  
</script>

**src\components\TheWelcome.vue**

<script setup lang="ts">  
import WelcomeItem from './WelcomeItem.vue'  
import DocumentationIcon from './icons/IconDocumentation.vue'  
import ToolingIcon from './icons/IconTooling.vue'  
import EcosystemIcon from './icons/IconEcosystem.vue'  
import CommunityIcon from './icons/IconCommunity.vue'  
import SupportIcon from './icons/IconSupport.vue'  
</script>  
  
<template>  
 <WelcomeItem>  
 <template #icon>  
 <DocumentationIcon />  
 </template>  
 <template #heading>Documentation</template>  
  
 Vue’s  
 <a href="https://vuejs.org/" target="\_blank" rel="noopener">official documentation</a>  
 provides you with all information you need to get started.  
 </WelcomeItem>  
  
 <WelcomeItem>  
 <template #icon>  
 <ToolingIcon />  
 </template>  
 <template #heading>Tooling</template>  
  
 This project is served and bundled with  
 <a href="https://vitejs.dev/guide/features.html" target="\_blank" rel="noopener">Vite</a>. The  
 recommended IDE setup is  
 <a href="https://code.visualstudio.com/" target="\_blank" rel="noopener">VSCode</a> +  
 <a href="https://github.com/johnsoncodehk/volar" target="\_blank" rel="noopener">Volar</a>. If  
 you need to test your components and web pages, check out  
 <a href="https://www.cypress.io/" target="\_blank" rel="noopener">Cypress</a> and  
 <a href="https://on.cypress.io/component" target="\_blank">Cypress Component Testing</a>.  
  
 <br />  
  
 More instructions are available in <code>README.md</code>.  
 </WelcomeItem>  
  
 <WelcomeItem>  
 <template #icon>  
 <EcosystemIcon />  
 </template>  
 <template #heading>Ecosystem</template>  
  
 Get official tools and libraries for your project:  
 <a href="https://pinia.vuejs.org/" target="\_blank" rel="noopener">Pinia</a>,  
 <a href="https://router.vuejs.org/" target="\_blank" rel="noopener">Vue Router</a>,  
 <a href="https://test-utils.vuejs.org/" target="\_blank" rel="noopener">Vue Test Utils</a>, and  
 <a href="https://github.com/vuejs/devtools" target="\_blank" rel="noopener">Vue Dev Tools</a>. If  
 you need more resources, we suggest paying  
 <a href="https://github.com/vuejs/awesome-vue" target="\_blank" rel="noopener">Awesome Vue</a>  
 a visit.  
 </WelcomeItem>  
  
 <WelcomeItem>  
 <template #icon>  
 <CommunityIcon />  
 </template>  
 <template #heading>Community</template>  
  
 Got stuck? Ask your question on  
 <a href="https://chat.vuejs.org" target="\_blank" rel="noopener">Vue Land</a>, our official  
 Discord server, or  
 <a href="https://stackoverflow.com/questions/tagged/vue.js" target="\_blank" rel="noopener"  
 >StackOverflow</a  
 >. You should also subscribe to  
 <a href="https://news.vuejs.org" target="\_blank" rel="noopener">our mailing list</a> and follow  
 the official  
 <a href="https://twitter.com/vuejs" target="\_blank" rel="noopener">@vuejs</a>  
 twitter account for latest news in the Vue world.  
 </WelcomeItem>  
  
 <WelcomeItem>  
 <template #icon>  
 <SupportIcon />  
 </template>  
 <template #heading>Support Vue</template>  
  
 As an independent project, Vue relies on community backing for its sustainability. You can help  
 us by  
 <a href="https://vuejs.org/sponsor/" target="\_blank" rel="noopener">becoming a sponsor</a>.  
 </WelcomeItem>  
</template>

**src\components\WelcomeItem.vue**

<template>  
 <div class="item">  
 <i>  
 <slot name="icon"></slot>  
 </i>  
 <div class="details">  
 <h3>  
 <slot name="heading"></slot>  
 </h3>  
 <slot></slot>  
 </div>  
 </div>  
</template>  
  
<style scoped>  
.item {  
 margin-top: 2rem;  
 display: flex;  
}  
  
.details {  
 flex: 1;  
 margin-left: 1rem;  
}  
  
i {  
 display: flex;  
 place-items: center;  
 place-content: center;  
 width: 32px;  
 height: 32px;  
  
 color: var(--color-text);  
}  
  
h3 {  
 font-size: 1.2rem;  
 font-weight: 500;  
 margin-bottom: 0.4rem;  
 color: var(--color-heading);  
}  
  
@media (min-width: 1024px) {  
 .item {  
 margin-top: 0;  
 padding: 0.4rem 0 1rem calc(var(--section-gap) / 2);  
 }  
  
 i {  
 top: calc(50% - 25px);  
 left: -26px;  
 position: absolute;  
 border: 1px solid var(--color-border);  
 background: var(--color-background);  
 border-radius: 8px;  
 width: 50px;  
 height: 50px;  
 }  
  
 .item:before {  
 content: ' ';  
 border-left: 1px solid var(--color-border);  
 position: absolute;  
 left: 0;  
 bottom: calc(50% + 25px);  
 height: calc(50% - 25px);  
 }  
  
 .item:after {  
 content: ' ';  
 border-left: 1px solid var(--color-border);  
 position: absolute;  
 left: 0;  
 top: calc(50% + 25px);  
 height: calc(50% - 25px);  
 }  
  
 .item:first-of-type:before {  
 display: none;  
 }  
  
 .item:last-of-type:after {  
 display: none;  
 }  
}  
</style>

**src\components\icons\IconCommunity.vue**

<template>  
 <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor">  
 <path  
 d="M15 4a1 1 0 1 0 0 2V4zm0 11v-1a1 1 0 0 0-1 1h1zm0 4l-.707.707A1 1 0 0 0 16 19h-1zm-4-4l.707-.707A1 1 0 0 0 11 14v1zm-4.707-1.293a1 1 0 0 0-1.414 1.414l1.414-1.414zm-.707.707l-.707-.707.707.707zM9 11v-1a1 1 0 0 0-.707.293L9 11zm-4 0h1a1 1 0 0 0-1-1v1zm0 4H4a1 1 0 0 0 1.707.707L5 15zm10-9h2V4h-2v2zm2 0a1 1 0 0 1 1 1h2a3 3 0 0 0-3-3v2zm1 1v6h2V7h-2zm0 6a1 1 0 0 1-1 1v2a3 3 0 0 0 3-3h-2zm-1 1h-2v2h2v-2zm-3 1v4h2v-4h-2zm1.707 3.293l-4-4-1.414 1.414 4 4 1.414-1.414zM11 14H7v2h4v-2zm-4 0c-.276 0-.525-.111-.707-.293l-1.414 1.414C5.42 15.663 6.172 16 7 16v-2zm-.707 1.121l3.414-3.414-1.414-1.414-3.414 3.414 1.414 1.414zM9 12h4v-2H9v2zm4 0a3 3 0 0 0 3-3h-2a1 1 0 0 1-1 1v2zm3-3V3h-2v6h2zm0-6a3 3 0 0 0-3-3v2a1 1 0 0 1 1 1h2zm-3-3H3v2h10V0zM3 0a3 3 0 0 0-3 3h2a1 1 0 0 1 1-1V0zM0 3v6h2V3H0zm0 6a3 3 0 0 0 3 3v-2a1 1 0 0 1-1-1H0zm3 3h2v-2H3v2zm1-1v4h2v-4H4zm1.707 4.707l.586-.586-1.414-1.414-.586.586 1.414 1.414z"  
 />  
 </svg>  
</template>

**src\components\icons\IconDocumentation.vue**

<template>  
 <svg xmlns="http://www.w3.org/2000/svg" width="20" height="17" fill="currentColor">  
 <path  
 d="M11 2.253a1 1 0 1 0-2 0h2zm-2 13a1 1 0 1 0 2 0H9zm.447-12.167a1 1 0 1 0 1.107-1.666L9.447 3.086zM1 2.253L.447 1.42A1 1 0 0 0 0 2.253h1zm0 13H0a1 1 0 0 0 1.553.833L1 15.253zm8.447.833a1 1 0 1 0 1.107-1.666l-1.107 1.666zm0-14.666a1 1 0 1 0 1.107 1.666L9.447 1.42zM19 2.253h1a1 1 0 0 0-.447-.833L19 2.253zm0 13l-.553.833A1 1 0 0 0 20 15.253h-1zm-9.553-.833a1 1 0 1 0 1.107 1.666L9.447 14.42zM9 2.253v13h2v-13H9zm1.553-.833C9.203.523 7.42 0 5.5 0v2c1.572 0 2.961.431 3.947 1.086l1.107-1.666zM5.5 0C3.58 0 1.797.523.447 1.42l1.107 1.666C2.539 2.431 3.928 2 5.5 2V0zM0 2.253v13h2v-13H0zm1.553 13.833C2.539 15.431 3.928 15 5.5 15v-2c-1.92 0-3.703.523-5.053 1.42l1.107 1.666zM5.5 15c1.572 0 2.961.431 3.947 1.086l1.107-1.666C9.203 13.523 7.42 13 5.5 13v2zm5.053-11.914C11.539 2.431 12.928 2 14.5 2V0c-1.92 0-3.703.523-5.053 1.42l1.107 1.666zM14.5 2c1.573 0 2.961.431 3.947 1.086l1.107-1.666C18.203.523 16.421 0 14.5 0v2zm3.5.253v13h2v-13h-2zm1.553 12.167C18.203 13.523 16.421 13 14.5 13v2c1.573 0 2.961.431 3.947 1.086l1.107-1.666zM14.5 13c-1.92 0-3.703.523-5.053 1.42l1.107 1.666C11.539 15.431 12.928 15 14.5 15v-2z"  
 />  
 </svg>  
</template>

**src\components\icons\IconEcosystem.vue**

<template>  
 <svg xmlns="http://www.w3.org/2000/svg" width="18" height="20" fill="currentColor">  
 <path  
 d="M11.447 8.894a1 1 0 1 0-.894-1.789l.894 1.789zm-2.894-.789a1 1 0 1 0 .894 1.789l-.894-1.789zm0 1.789a1 1 0 1 0 .894-1.789l-.894 1.789zM7.447 7.106a1 1 0 1 0-.894 1.789l.894-1.789zM10 9a1 1 0 1 0-2 0h2zm-2 2.5a1 1 0 1 0 2 0H8zm9.447-5.606a1 1 0 1 0-.894-1.789l.894 1.789zm-2.894-.789a1 1 0 1 0 .894 1.789l-.894-1.789zm2 .789a1 1 0 1 0 .894-1.789l-.894 1.789zm-1.106-2.789a1 1 0 1 0-.894 1.789l.894-1.789zM18 5a1 1 0 1 0-2 0h2zm-2 2.5a1 1 0 1 0 2 0h-2zm-5.447-4.606a1 1 0 1 0 .894-1.789l-.894 1.789zM9 1l.447-.894a1 1 0 0 0-.894 0L9 1zm-2.447.106a1 1 0 1 0 .894 1.789l-.894-1.789zm-6 3a1 1 0 1 0 .894 1.789L.553 4.106zm2.894.789a1 1 0 1 0-.894-1.789l.894 1.789zm-2-.789a1 1 0 1 0-.894 1.789l.894-1.789zm1.106 2.789a1 1 0 1 0 .894-1.789l-.894 1.789zM2 5a1 1 0 1 0-2 0h2zM0 7.5a1 1 0 1 0 2 0H0zm8.553 12.394a1 1 0 1 0 .894-1.789l-.894 1.789zm-1.106-2.789a1 1 0 1 0-.894 1.789l.894-1.789zm1.106 1a1 1 0 1 0 .894 1.789l-.894-1.789zm2.894.789a1 1 0 1 0-.894-1.789l.894 1.789zM8 19a1 1 0 1 0 2 0H8zm2-2.5a1 1 0 1 0-2 0h2zm-7.447.394a1 1 0 1 0 .894-1.789l-.894 1.789zM1 15H0a1 1 0 0 0 .553.894L1 15zm1-2.5a1 1 0 1 0-2 0h2zm12.553 2.606a1 1 0 1 0 .894 1.789l-.894-1.789zM17 15l.447.894A1 1 0 0 0 18 15h-1zm1-2.5a1 1 0 1 0-2 0h2zm-7.447-5.394l-2 1 .894 1.789 2-1-.894-1.789zm-1.106 1l-2-1-.894 1.789 2 1 .894-1.789zM8 9v2.5h2V9H8zm8.553-4.894l-2 1 .894 1.789 2-1-.894-1.789zm.894 0l-2-1-.894 1.789 2 1 .894-1.789zM16 5v2.5h2V5h-2zm-4.553-3.894l-2-1-.894 1.789 2 1 .894-1.789zm-2.894-1l-2 1 .894 1.789 2-1L8.553.106zM1.447 5.894l2-1-.894-1.789-2 1 .894 1.789zm-.894 0l2 1 .894-1.789-2-1-.894 1.789zM0 5v2.5h2V5H0zm9.447 13.106l-2-1-.894 1.789 2 1 .894-1.789zm0 1.789l2-1-.894-1.789-2 1 .894 1.789zM10 19v-2.5H8V19h2zm-6.553-3.894l-2-1-.894 1.789 2 1 .894-1.789zM2 15v-2.5H0V15h2zm13.447 1.894l2-1-.894-1.789-2 1 .894 1.789zM18 15v-2.5h-2V15h2z"  
 />  
 </svg>  
</template>

**src\components\icons\IconSupport.vue**

<template>  
 <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" fill="currentColor">  
 <path  
 d="M10 3.22l-.61-.6a5.5 5.5 0 0 0-7.666.105 5.5 5.5 0 0 0-.114 7.665L10 18.78l8.39-8.4a5.5 5.5 0 0 0-.114-7.665 5.5 5.5 0 0 0-7.666-.105l-.61.61z"  
 />  
 </svg>  
</template>

**src\components\icons\IconTooling.vue**

<!-- This icon is from <https://github.com/Templarian/MaterialDesign>, distributed under Apache 2.0 (https://www.apache.org/licenses/LICENSE-2.0) license-->  
<template>  
 <svg  
 xmlns="http://www.w3.org/2000/svg"  
 xmlns:xlink="http://www.w3.org/1999/xlink"  
 aria-hidden="true"  
 role="img"  
 class="iconify iconify--mdi"  
 width="24"  
 height="24"  
 preserveAspectRatio="xMidYMid meet"  
 viewBox="0 0 24 24"  
 >  
 <path  
 d="M20 18v-4h-3v1h-2v-1H9v1H7v-1H4v4h16M6.33 8l-1.74 4H7v-1h2v1h6v-1h2v1h2.41l-1.74-4H6.33M9 5v1h6V5H9m12.84 7.61c.1.22.16.48.16.8V18c0 .53-.21 1-.6 1.41c-.4.4-.85.59-1.4.59H4c-.55 0-1-.19-1.4-.59C2.21 19 2 18.53 2 18v-4.59c0-.32.06-.58.16-.8L4.5 7.22C4.84 6.41 5.45 6 6.33 6H7V5c0-.55.18-1 .57-1.41C7.96 3.2 8.44 3 9 3h6c.56 0 1.04.2 1.43.59c.39.41.57.86.57 1.41v1h.67c.88 0 1.49.41 1.83 1.22l2.34 5.39z"  
 fill="currentColor"  
 ></path>  
 </svg>  
</template>

**src\views\AboutView.vue**

<template>  
 <div class="about">  
 <h1>This is an about page</h1>  
 </div>  
</template>  
  
<style>  
@media (min-width: 1024px) {  
 .about {  
 min-height: 100vh;  
 display: flex;  
 align-items: center;  
 }  
}  
</style>

**src\views\AddRecordView.vue**

<template>  
 添加数据记录  
</template>  
  
<script setup lang="ts">  
</script>

**src\views\DataView.vue**

<template>  
 <el-card class="guest-container">  
 <template #header>  
 <div class="header">  
 <el-button type="primary" size="small" icon="el-icon-plus" @click="handleSolve">修改数据记录</el-button>  
 <el-button type="danger" size="small" icon="el-icon-delete" @click="handleForbid">删除数据记录</el-button>  
 </div>  
 </template>  
 <Table action='/environmentData/pageQueryOrderByTime' ref="table">  
 <template #column>  
 <el-table-column type="selection" width="55">  
 </el-table-column>  
 <el-table-column prop="time" label="测量时间">  
 </el-table-column>  
 <el-table-column prop="lightIntensity" label="光照强度">  
 </el-table-column>  
 <el-table-column prop="temperature" label="温度">  
 </el-table-column>  
 <el-table-column prop="airHumidity" label="湿度">  
 </el-table-column>  
 <el-table-column prop="soilMoisture" label="土壤湿度">  
 </el-table-column>  
 <el-table-column label="是否删除">  
 <template #default="scope">  
 <span :style="scope.row.lockedFlag == 0 ? 'color: green;' : 'color: red;'">  
 {{ scope.row.lockedFlag == 0 ? '调整' : '删除' }}  
 </span>  
 </template>  
 </el-table-column>  
 <!-- <el-table-column prop="createTime" label="注册时间">  
 </el-table-column> -->  
 </template>  
 </Table>  
 </el-card>  
</template>  
  
<script>  
import { ref } from 'vue'  
import Table from '@/components/Table.vue'  
import { ElMessage } from 'element-plus'  
import { Plus, Delete } from '@element-plus/icons-vue'  
import axios from 'axios'  
  
const table = ref(null) // 绑定 Table 的 ref 属性  
// 解禁方法  
const handleSolve = () => {  
 // 可以通过 table 获取到组件内部的 multipleSelection 值  
 if (!table.value.state.multipleSelection.length) {  
 ElMessage.error('请选择项')  
 return  
 }  
 axios.put(`/users/0`, {  
 ids: table.value.state.multipleSelection.map(item => item.userId)  
 }).then(() => {  
 ElMessage.success('解除成功')  
 // 通过 table 获取组件内部的 getList 方法  
 table.value.getList()  
 })  
}  
// 禁用方法  
const handleForbid = () => {  
 // 可以通过 table 获取到组件内部的 multipleSelection 值  
 if (!table.value.state.multipleSelection.length) {  
 ElMessage.error('请选择项')  
 return  
 }  
 axios.put(`/users/1`, {  
 ids: table.value.state.multipleSelection.map(item => item.userId)  
 }).then(() => {  
 ElMessage.success('禁用成功')  
 // 通过 table 获取组件内部的 getList 方法  
 table.value.getList()  
 })  
}  
</script>

**src\views\envData.vue**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>环境监测系统展示</title>  
 <link rel="stylesheet" href="./css/envData.css">  
 <script src="js/jquery.min.js"></script>  
 <script src="js/axios.min.js"></script>  
</head>  
<body>  
 <!-- 导航栏 -->  
 <div class="nav">  
 <div id="nav-left">  
 <img src="./img/yhtx02.png" alt="" id="logo">  
 <span>&nbsp;&nbsp;&nbsp;环境检测系统</span>  
 </div>  
 <div id="nav-right">  
 <a href="./envLogin.html">注销</a>  
 </div>  
 </div>  
</body>  
</html>

**src\views\IndexView.vue**

<script setup lang="ts">  
import axios from 'axios';  
import { reactive } from 'vue'  
import { dataType } from 'element-plus/es/components/table-v2/src/common';  
import TheWelcome from '../components/TheWelcome.vue'  
// const ENV = import.meta.env  
// export default {  
// name: 'HomeView',  
// setup() {  
// console.log(ENV)  
// }  
// }  
// 将上述代码转化为setup模式  
const ENV = import.meta.env  
console.log(ENV)  
import { onMounted, onUnmounted } from 'vue'  
  
const state = reactive({  
 total: 0,  
 today: 0,  
 runningTime: 0  
})  
/\*\*  
 \* 时间戳转换为时间函数  
 \*/  
function timestampToTime(timestamp) {  
 timestamp = timestamp ? timestamp : null;  
 let date = new Date(timestamp);//时间戳为10位需\*1000，时间戳为13位的话不需乘1000  
 let Y = date.getFullYear() + '-';  
 let M = (date.getMonth() + 1 < 10 ? '0' + (date.getMonth() + 1) : date.getMonth() + 1) + '-';  
 let D = (date.getDate() < 10 ? '0' + date.getDate() : date.getDate()) + ' ';  
 let h = (date.getHours() < 10 ? '0' + date.getHours() : date.getHours()) + ':';  
 let m = (date.getMinutes() < 10 ? '0' + date.getMinutes() : date.getMinutes()) + ':';  
 let s = date.getSeconds() < 10 ? '0' + date.getSeconds() : date.getSeconds();  
 return Y + M + D + h + m + s;  
}  
  
  
  
let myChart = null  
  
onMounted(() => {  
 if (window.echarts) {  
 // 基于准备好的dom，初始化echarts实例  
 myChart = window.echarts.init(document.getElementById('zoom'))  
  
 axios.get('/api/environmentData/all').then(res => {  
 console.log(res.data)  
 state.total = res.data.length  
 // 指定图表的配置项和数据  
 const option = {  
 title: {  
 text: '系统折线图'  
 },  
 tooltip: {  
 trigger: 'axis',  
 axisPointer: {  
 type: 'cross',  
 label: {  
 backgroundColor: '#6a7985'  
 }  
 }  
 },  
 legend: {  
 data: ['光照强度', '温度', '湿度', '土壤湿度']  
 },  
 toolbox: {  
 feature: {  
 saveAsImage: {}  
 }  
 },  
 grid: {  
 left: '3%',  
 right: '4%',  
 bottom: '3%',  
 containLabel: true  
 },  
 xAxis: [  
 {  
 type: 'category',  
 boundaryGap: false,  
 // data: ['2021-03-11', '2021-03-12', '2021-03-13', '2021-03-14', '2021-03-15', '2021-03-16', '2021-03-17']  
 data: res.data.map(item => item.time)  
 }  
 ],  
 yAxis: [  
 {  
 type: 'value'  
 }  
 ],  
 series: [  
 {  
 name: '光照强度',  
 type: 'line',  
 areaStyle: {},  
 emphasis: {  
 focus: 'series'  
 },  
 // data: [120, 132, 101, 134, 90, 230, 210]  
 data: res.data.map(item => item.lightIntensity)  
 },  
 {  
 name: '温度',  
 type: 'line',  
 areaStyle: {},  
 emphasis: {  
 focus: 'series'  
 },  
 // data: [220, 182, 191, 234, 290, 330, 310]  
 data: res.data.map(item => item.temperature)  
 },  
 {  
 name: '湿度',  
 type: 'line',  
 areaStyle: {},  
 emphasis: {  
 focus: 'series'  
 },  
 // data: [150, 232, 201, 154, 190, 330, 410]  
 data: res.data.map(item => item.airHumidity)  
 },  
 {  
 name: '土壤湿度',  
 type: 'line',  
 areaStyle: {},  
 emphasis: {  
 focus: 'series'  
 },  
 // data: [320, 332, 301, 334, 390, 330, 320]  
 data: res.data.map(item => item.soilMoisture)  
 }  
 ]  
 }  
  
 // 使用刚指定的配置项和数据显示图表。  
 myChart.setOption(option)  
  
 var today = new Date();  
 var todaynum = Date.parse(new Date(today.toLocaleDateString()).toString());  
 var min = null  
 var max = null  
 for (let i = 0; i < res.data.length; i++) {  
 if (Date.parse(new Date(res.data[i].time).toString()) - todaynum >= 0) {  
 state.today += 1;  
 }  
 // 找到第一个不为空的时间赋值给min和max  
 if (min == null && res.data[i].time != null) {  
 min = res.data[i].time  
 }  
 if (max == null && res.data[i].time != null) {  
 max = res.data[i].time  
 }  
 // 找最大最小值  
 for (let j = 0; j < res.data.length; j++) {  
 if ((Date.parse(new Date(res.data[i].time).toString()) < Date.parse(new Date(min).toString())) && res.data[i].time != null) {  
 min = res.data[i].time  
 }  
 if (Date.parse(new Date(res.data[i].time).toString()) > Date.parse(new Date(max).toString())) {  
 max = res.data[i].time  
 }  
 }  
 }  
 console.log(min, max)  
 // 求出最大最小时间戳的差转化为  
 state.runningTime = timestampToTime((Date.parse(new Date(max).toString()) - Date.parse(new Date(min).toString())))  
 })  
  
 }  
})  
onUnmounted(() => {  
 myChart.dispose()  
})  
  
</script>  
  
<template>  
 <el-card class="introduce">  
 <div class="order">  
 <el-card class="order-item">  
 <template #header>  
 <div class="card-header">  
 <span>历史记录总数</span>  
 </div>  
 </template>  
 <div class="item">{{ state.total }}</div>  
 </el-card>  
 <el-card class="order-item">  
 <template #header>  
 <div class="card-header">  
 <span>今日数据</span>  
 </div>  
 </template>  
 <div class="item">{{ state.today }}</div>  
 </el-card>  
 <el-card class="order-item">  
 <template #header>  
 <div class="card-header">  
 <span>连续运行时间</span>  
 </div>  
 </template>  
 <div class="item">{{ state.runningTime }}</div>  
 </el-card>  
 </div>  
 <div id="zoom"></div>  
 </el-card>  
</template>  
  
  
<style>  
.introduce .order {  
 display: flex;  
 margin-bottom: 50px;  
}  
  
.introduce .order .order-item {  
 flex: 1;  
 margin-right: 20px;  
}  
  
.introduce .order .order-item:last-child {  
 margin-right: 0;  
}  
  
#zoom {  
 min-height: 500px;  
}  
</style>

**src\views\LoginView.vue**

<script setup lang="ts">  
// import axios from '@/utils/axios'  
import axios from 'axios'  
// 安装 js-md5，密码需要 md5 加密，服务端是解密 md5 的形式  
import md5 from 'js-md5'  
import { reactive, ref, toRefs } from 'vue'  
import { localSet } from '@/utils'  
import { createRequire } from 'module';  
  
// el-form 组件接收一个 ref 属性，Vue3 需要这样声明  
const loginForm = ref(null)  
const state = reactive({  
 ruleForm: {  
 username: '', // 账号  
 password: '', // 密码  
 },  
 checked: true,  
 // 表单验证判断。  
 rules: {  
 username: [  
 { required: 'true', message: '账户不能为空', trigger: 'blur' }  
 ],  
 password: [  
 { required: 'true', message: '密码不能为空', trigger: 'blur' }  
 ]  
 }  
})  
// 表单提交方法  
const submitForm = async () => {  
 loginForm.value.validate((valid) => {  
 // valid 是一个布尔值，表示表单是否通过了上面 rules 的规则。  
 if (valid) {  
 // // /adminUser/login 登录接口路径  
 // axios.post('/user/login', {  
 // userName: state.ruleForm.username || '',  
 // passwordMd5: md5(state.ruleForm.password), // 密码需要 md5 加密  
 // }).then(res => {  
 // // 返回的时候会有一个 token，这个令牌就是我们后续去请求别的接口时要带上的，否则会报错，非管理员。  
 // // 这里我们将其存储到 localStorage 里面。  
 // localSet('token', res)  
 // // 此处登录完成之后，需要刷新页面  
 // window.location.href = '/'  
 // })  
  
  
  
 // 硬编码的  
 var config = {  
 method: 'post',  
 url: '/api/user/login?username=' + state.ruleForm.username + '&password=' + state.ruleForm.password,  
 headers: {}  
 };  
  
 axios(config)  
 .then(function (response) {  
 console.log(JSON.stringify(response.data));  
 if (response.data == "") {  
 ElMessage.error('登陆失败');  
 } else {  
 //记录id  
 localSet('id', response.data.id)  
 localSet('userInfo', response.data)  
 // 此处登录完成之后，需要刷新页面  
 window.location.href = '/'  
 }  
  
 })  
 .catch(function (error) {  
 console.log(error);  
 });  
  
  
  
 } else {  
 console.log('error submit!!')  
 return false;  
 }  
 })  
}  
// 重制方法  
const resetForm = () => {  
 // loginForm能拿到 el-form 的重制方法  
 loginForm.value.resetFields();  
}  
</script>  
  
<template>  
 <div class="login-body">  
 <!--登录框div-->  
 <div class="login-container">  
 <!--登录框头部logo部分-->  
 <div class="head">  
 <!-- <img class="logo" src="https://s.weituibao.com/1582958061265/mlogo.png" /> -->  
 <div class="name">  
 <div class="title">植物生长环境监测系统</div>  
 <div class="tips">SpringBoot + Vue3.0</div>  
 </div>  
 </div>  
 <!--loginForm是从setup内返回得到的-->  
 <el-form label-position="top" :rules="state.rules" :model="state.ruleForm" ref="loginForm"  
 class="login-form">  
 <el-form-item label="账号" prop="username">  
 <el-input type="text" v-model.trim="state.ruleForm.username" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item label="密码" prop="password">  
 <el-input type="password" v-model.trim="state.ruleForm.password" autocomplete="off"></el-input>  
 </el-form-item>  
 <el-form-item>  
 <div style="color: #333">登录表示您已同意<a>《服务条款》</a></div>  
 <el-button style="width: 100%" type="primary" @click="submitForm">立即登录</el-button>  
 <el-checkbox v-model="checked" @change="!checked">下次自动登录</el-checkbox>  
 </el-form-item>  
 </el-form>  
 </div>  
 </div>  
</template>  
  
<style scoped>  
.login-body {  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 width: 100%;  
 background-color: #fff;  
 background-image: url(../assets/img2.png);  
}  
  
.login-container {  
 width: 420px;  
 height: 500px;  
 background-color: #fff;  
 border-radius: 4px;  
 box-shadow: 0px 21px 41px 0px rgba(0, 0, 0, 0.2);  
}  
  
.head {  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 padding: 40px 0 20px 0;  
}  
  
.head img {  
 width: 100px;  
 height: 100px;  
 margin-right: 20px;  
}  
  
.head .title {  
 font-size: 28px;  
 color: #1BAEAE;  
 font-weight: bold;  
}  
  
.head .tips {  
 font-size: 12px;  
 color: #999;  
}  
  
.login-form {  
 width: 70%;  
 margin: 0 auto;  
}  
  
.login-form>>>.el-form--label-top .el-form-item\_\_label {  
 padding: 0;  
}  
  
.login-form>>>.el-form-item {  
 margin-bottom: 0;  
}  
</style>