第13章 Spring Boot+Vue 前后端分离项目实战

13.2 组合式 API 项目实战

组合式 API 是 setup 语法糖的一个过渡，单纯使用组合式 API 并没有减少太多代码的开发，但因为 setup 语法糖过于简略，并不合适初学者，所以本节使用组合式 API 对 Alan 人事管理系统进行改造，通过该“过渡”更好地在下一节使用 setup 语法糖进行实战。

复制项目 hrsys_options 并重命名为 hrsys_composition。

因为使用组合式 API 无须额外进行环境配置，所以无须改变 axios、Bootstrap、路由的配置。

无论是组合式 API 还是 setup 语法糖，都是对 Vue 组件的 TypeScript 代码进行改造，**并不涉及 < template > 和 < style > 标签里的内容。**

对 Index.vue 组件的改造如下:

```
<script lang="ts">
import {onMounted} from "vue";
import {useRouter} from "vue-router";

export default {
setup(){
   const router = useRouter();
  onMounted(()=>{
    router.push("/emp/show");
  })
},
}
</script>
```

观察以上代码，由于在 setup （）方法中无法使用 this，所以不能像选项式 API 那样通过this.$router 的方式获取路由对象，Vue 3 提供了通过 useRouter （）方法获取路由对象的功能。

对 Employee 组件show.vue的改造如下:

```
<script>
import axios from "../../util/axiosInstance"
import {onMounted, ref,reactive} from "vue";
import {useRouter} from "vue-router";

export default {

  setup() {
    const router = useRouter();
    const selectedId = ref(-1);
    const datas=reactive({
     form :{
       number: null,
       name: null,
       gender: "",
    age: null,
    'dep.id': ""
  },
  list:[],
  depList:[]
})

 const search=()=> {
   axios.get('/emp', {params: datas.form})
       .then((res)=> {
         datas.list=res.data ;

       })
 }
 const searchDep =()=>{
   axios.get('/dep')
       .then((res)=> {
         datas.depList=res.data ;

       })
 }
 const selectTr =(id)=> {
   console.log(id)
  selectedId.value = id;

 }
 const showAdd=()=> {
   router.push({name: "EmpAdd"})
 }
 const showUpdate =()=> {
   if (selectedId.value > -1) {
     router.push({name: "EmpUpdate", query: {id: selectedId.value}})
   } else {
     alert("请选中数据");
   }
 }

 const deleteData =()=>{
   if (selectedId.value > -1) {
     axios.delete('/emp/' + selectedId.value)
         .then( (res)=> {
           if (res.data == true) {
            search();
           } else {
             alert("系统错误");
           }
         })
   } else {
     alert("请选中数据");
   }
 }


 onMounted(() => {
   search();
   searchDep();

 })
```

```
  return {
   selectedId, datas, search, searchDep, selectTr, showAdd, showUpdate,
deleteData
  }
 }
}
</script>
```

巧妙地使用 **reactive（）函数**是简化响应式对象操作的关键，如在以上代码中，定义**响应式常量 datas**，但 datas 对象内部定义的 form 属性、list 属性才是 < template > 标签中真正被使用的数据，所以不需要依赖 Vue 3 提供的复杂 API 操作来改变响应式常量的值，直接通过datas.form 形式赋值即可。

当然，这样的设计需要在模板中分别以 datas.form 和 datas.list 的方式使用 form 和 list。

```
<select name="gender" class="form-control" v-model="datas.form.gender">
      <option value="">性别</option>
      <option value="男">男</option>
      <option value="女">女</option>
</select>
```

show.vue 完整代码如下：

```
<script>
import axios from "../../util/axiosInstance"
import {onMounted, ref,reactive} from "vue";
import {useRouter} from "vue-router";

export default {

 setup() {
  const router = useRouter();
  const selectedId = ref(-1);
  const datas=reactive({
   form :{
    number: null,
    name: null,
    gender: "",
    age: null,
    'dep.id': ""
   },
   list:[],
   depList:[]
  })

  const search=()=> {
   axios.get('/emp', {params: datas.form})
     .then((res)=> {
      datas.list=res.data ;

     })
  }
  const searchDep =()=>{
   axios.get('/dep')
     .then((res)=> {
      datas.depList=res.data ;
```

```
          })
    }
    const selectTr =(id)=> {
      console.log(id)
     selectedId.value = id;

    }
    const showAdd=()=> {
      router.push({name: "EmpAdd"})
    }
    const showUpdate =()=> {
      if (selectedId.value > -1) {
        router.push({name: "EmpUpdate", query: {id: selectedId.value}})
      } else {
        alert("请选中数据");
      }
    }

    const deleteData =()=>{
      if (selectedId.value > -1) {
        axios.delete('/emp/' + selectedId.value)
            .then( (res)=> {
              if (res.data == true) {
               search();
              } else {
                alert("系统错误");
              }
            })
      } else {
        alert("请选中数据");
      }
    }


    onMounted(() => {
      search();
      searchDep();

    })
    return {
      selectedId, datas, search, searchDep, selectTr, showAdd, showUpdate,
deleteData
    }
  }
}
</script>

<template>
  <div id="container">
    <div class="form-horizontal">
      <form class="form-group" @submit.prevent="search()">
        <div class="col-sm-2">
          <input type="text" class="form-control"
                 placeholder="编号" v-model="datas.form.number">
        </div>
        <div class="col-sm-2">
          <input type="text" class="form-control"
```

```
                 placeholder="姓名" v-model="datas.form.name">
        </div>
        <div class="col-sm-2 ">
          <select name="gender" class="form-control" v-
model="datas.form.gender">
            <option value="">性别</option>
            <option value="男">男</option>
            <option value="女">女</option>
          </select>
        </div>
        <div class="col-sm-2">
          <input type="text" class="form-control" placeholder="年龄" v-
model="datas.form.age">
        </div>
        <div class="col-sm-2 ">
          <select class="form-control" v-model="datas.form['dep.id']">
            <option value="">部门</option>
            <option v-for="dep in datas.depList"
                    v-text="dep.name" v-bind:key="dep.id" v-bind:value="dep.id">
            </option>
          </select>
        </div>
        <div class="col-sm-2">
          <button class="btn btn-primary">搜索</button>
        </div>
      </form>
    </div>
    <table class="table table-striped table-bordered table-hover">
      <thead>
      <tr>
        <th>编号</th>
        <th>名字</th>
        <th>性别</th>
        <th>年龄</th>
        <th>部门</th>
      </tr>
      </thead>
      <tbody class="scrollable-tbody">
      <tr class="data" v-for="emp in datas.list" v-bind:key="emp.id" v-
bind:class="{selected:emp.id==selectedId}"
          @click="selectTr(emp.id)">
        <td v-text="emp.number"></td>
        <td v-text="emp.name"></td>
        <td v-text="emp.gender"></td>
        <td v-text="emp.age"></td>
        <td v-text="emp.dep!=null?emp.dep.name:''"></td>
      </tr>
      </tbody>
    </table>
    <div id="buttons">
      <button type="button" class="btn btn-primary" @click="showAdd">新增
</button>
      <button type="button" class="btn btn-primary" @click="showUpdate">修改
</button>
      <button type="button" class="btn btn-danger" @click="deleteData">删除
</button>
    </div>
  </div>
```

```
</template>


<style scoped>
#container {
  width: 800px;
  margin: 10px auto;
}

#container .selected {
  background: #337ab7
}

#buttons button {
  margin-right: 5px;
}

 .table {
        width: 100%;
        display: block;
        overflow-x: auto;
    }

    .table thead,
    .table tbody tr {
        display: table;
        width: 100%;
        table-layout: fixed;
    }

    .table thead {
        width: calc(100% - 17px); /* Adjust for scrollbar width */
    }

    .scrollable-tbody {
        display: block;
        max-height: 300px; /* 根据需要调整高度 */
        overflow-y: auto;
        width: 100%;
    }

</style>
```

Add.vue完整代码如下：

```
<script>
import axios from "../../util/axiosInstance"
import {onMounted,  reactive} from "vue";
import {useRouter} from "vue-router";

export default {
  setup() {
    const router = useRouter();
    const datas = reactive({
      form:
          {
            number: null,
            name: null,
            gender: null,
            age: null,
            dep: {id: ""}
          },
      depList: []
    })


    const add = () => {
      axios.post('/emp', datas.form)
          .then((res) => {
            if (res.data == true) {
              router.push({name: "EmpShow"})
            } else {
              alert("系统错误");
            }
          })
    }
    const searchDep = () => {
      axios.get('/dep')
          .then((res) => {
            datas.depList=res.data;
            datas.form.dep.id=res.data[0].id
          })
    }

    onMounted(() => {
      searchDep();
    })
    return {datas, add, searchDep}
  }
}
</script>

<template>
  <div id="container">
    <form class="form-horizontal" @submit.prevent="add">
      <div class="form-group">
        <label class="col-sm-2 control-label">编号</label>
        <div class="col-sm-10">
          <input type="text" class="form-control" placeholder="请输入编号" v-
model="datas.form.number">
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-2 control-label">名字</label>
        <div class="col-sm-10">
          <input type="text" class="form-control" placeholder="请输入名字"
                 v-model="datas.form.name">
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-2 control-label">性别</label>
        <div class="col-sm-10">
          <input type="radio" value="男" v-model="datas.form.gender"/>男<input
type="radio"
```

```
value="女" v-model="datas.form.gender"/>女
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-2 control-label">年龄</label>
        <div class="col-sm-10">
          <input type="text" class="form-control" placeholder="请输入年龄"
                 v-model="datas.form.age">
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-2 control-label">部门</label>
        <div class="col-sm-10">
          <select class="form-control" v-model="datas.form.dep.id">
            <option v-for="dep in datas.depList"
                    v-text="dep.name" v-bind:key="dep.id" v-bind:value="dep.id">
            </option>
          </select>
        </div>
      </div>
      <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
          <button type="submit" class="btn btn-primary">保存</button>
        </div>
      </div>
    </form>
  </div>
</template>


<style scoped>
#container {
  width: 600px;
  margin: 10px auto;
}
</style>
```

Update.vue完整代码如下：

```
<script>
import axios from "../../util/axiosInstance"
import {onMounted, ref, reactive} from "vue";
import {useRoute, useRouter} from "vue-router";

export default {
  setup() {
    const router = useRouter();
    const route = useRoute()
    const datas = reactive({
      form: {
        id: null,
        number: null,
        name: null,
        gender: null,
        age: null,
        dep: {id: ""}
      },
      depList: []
    })

    const update = () => {
      axios.put('/emp', datas.form)
          .then((res) => {
            if (res.data == true) {
              router.push({name: "EmpShow"})
            } else {
              alert("系统错误");
            }
          })
    }
    const searchById = () => {
      axios.get('/emp/' + datas.form.id)
          .then((res) => {
            datas.form= res.data;
            if (datas.form.dep == null) {
              datas.form.dep = {id: datas.depList[0].id};
            }
          })
    }
    const searchDep = () => {
      axios.get('/dep')
          .then((res) => {
            datas.depList=res.data;
            datas.form.dep.id = datas.depList[0].id

          })
    }


    datas.form.id = route.query.id;
    searchDep();
    searchById();


    return {datas, update, searchById, searchDep}
  }
}
</script>
<template>
  <div id="container">
    <form class="form-horizontal" @submit.prevent="update">
      <div class="form-group">
        <label class="col-sm-2 control-label">编号</label>
        <div class="col-sm-10">
          <input type="text" class="form-control" placeholder="请输入编号" v-
model="datas.form.number">
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-2 control-label">名字</label>
        <div class="col-sm-10">
          <input type="text" class="form-control" placeholder="请输入名字"
                 v-model="datas.form.name">
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-2 control-label">性别</label>
        <div class="col-sm-10">
          <input type="radio" value="男" v-model="datas.form.gender"/>男 <input
            type="radio" value="女" v-model="datas.form.gender"/>女
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-2 control-label">年龄</label>
        <div class="col-sm-10">
          <input type="text" class="form-control" placeholder="请输入年龄"
                 v-model="datas.form.age">
        </div>
      </div>
      <div class="form-group">
        <label class="col-sm-2 control-label">部门</label>
        <div class="col-sm-10">
          <select class="form-control" v-model="datas.form.dep.id">
            <option v-for="dep in datas.depList"
                    v-text="dep.name" v-bind:key="dep.id" v-bind:value="dep.id">
            </option>
          </select>
        </div>
      </div>
      <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
          <button type="submit" class="btn btn-primary">保存</button>
        </div>
      </div>
    </form>
  </div>
</template>


<style scoped>
#container {
  width: 600px;
  margin: 10px auto;
}
</style>
```