

第13章 Spring Boot+Vue 前后端分离项目实战

13.3 setup 语法糖项目实战

与组合式 API 方式相比，setup 语法糖带来的简便便是省略了定义导出对象、setup () 方法书写、return 响应式常量。虽然省略的代码不多，但这可以让开发者不用去关注需要 return 哪些响应式常量给模板使用，给实际项目开发带来了巨大的方便。

复制项目 hrsys_composition 并重命名为 hrsys_setup。

因为 <script> 标签中的代码是从上往下解释执行的，所以不需要在生命周期 onMounted () 方法中调用 search () 方法和 searchDep () 方法，直接调用它们即可。

show.vue代码如下：

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

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");
  }
}
search();
searchDep();
</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代码如下:

```

<script lang="ts" setup>
import axios from "../../util/axiosInstance"
import {reactive} from "vue";
import {useRouter} from "vue-router";

const router = useRouter();
const datas = reactive({
  form: {
    number: null,
    name: null,
    gender: null,
    age: null,
    dep: {id: null}
  },
  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
        console.log("==" +res.data[0].id)
        console.log("==" +datas.form.dep.id)
      })
  }

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代码如下:

```

<script lang="ts" setup>
import axios from "../../util/axiosInstance"
import { reactive } from "vue";
import { useRoute, useRouter } from "vue-router";

const router = useRouter();
const route = useRoute()
const datas = reactive({
  form: {
    id: null,
    number: null,
    name: null,
    gender: null,
    age: null,
    dep: {id: null}
  },
  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();

</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>

```

```
    </form>
  </div>
</template>
```

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