


```

    <el-descriptions-item label="姓名"> <el-button @click="unbindToken" link type="primary" >解绑
  </el-button></el-descriptions-item>
  </el-descriptions>
</div>
<div v-else style="padding-bottom:20px">
  <el-descriptions :column="1" border >
    <el-descriptions-item label="姓名"> <el-tag type="danger">危险</el-tag> </el-descriptions-
  item>
    <el-descriptions-item label="Token"> <el-input v-model="edittoken" placeholder="请输入Token"></el-input> </el-descriptions-item>
    <el-descriptions-item label="姓名"> <el-input v-model="editname" placeholder="请输入姓名"></el-
  input> </el-descriptions-item>
    <el-descriptions-item label="操作"> <el-button @click="bindToken" type="primary" >绑定</el-
  button></el-descriptions-item>
  </el-descriptions>
</div>

```

姓名 危险

- 姓名 危险 UI
- 姓名 危险 危险
- 姓名 危险 危险
- 姓名 危险

2.2.2 折叠面板

```

<el-collapse v-if="token" v-model="activeNames" @change="handleChange">
  <el-collapse-item title="姓名" name="1">
    <el-form-item label="姓名">
      <el-input v-model="buyer.name" :disabled="!buyer.edit" placeholder="请输入姓名"></el-input>
    </el-form-item>
    <el-form-item label="地址">
      <el-input v-model="buyer.address" :disabled="!buyer.edit" placeholder="请输入地址"></el-
  input>
    </el-form-item>
    <el-form-item label="手机">
      <el-input v-model="buyer.mobile" :disabled="!buyer.edit" placeholder="请输入手机"></el-input>
    </el-form-item>
    <el-form-item label="联系人">
      <el-input v-model="buyer.contact" :disabled="!buyer.edit" placeholder="请输入联系人"></el-
  input>

```

```

</el-form-item>
<div style=" margin-bottom:20px">
  <div v-if="token" >
    <el-button v-if="!buyer.edit" @click="buyer.edit=true" type="primary"></el-button>
    <el-button v-else @click="addBuyer" type="primary"></el-button>
  </div>
  <div v-else>
    <el-button @click="addBuyer" type="primary"></el-button>
  </div>
</div>
</el-collapse-item>
</el-collapse>

```

□□□ □

- □□□□□□□□□□□□□□□□
- □□□□ /□□□□□□
- □□□□□□□□□□□□□□□□

2.3 ??????????

2.3.1 ??????

```

<el-table :data="tableData" height="calc(100vh - 145px)" >
  <el-table-column prop="name" label="□□" >
    <template #default="scope">
      <div>{{scope.row.name}}</div>
      <div class="font-extraSmall">{{scope.row.address}}</div>
    </template>
  </el-table-column>
  <el-table-column prop="contact" label="□□□" width="230" >
    <template #default="scope">
      <div>{{scope.row.contact}}</div>
      <div>{{scope.row.mobile}}</div>
    </template>
  </el-table-column>
  <el-table-column prop="createtime" label="□□" width="240" v-if="isowner" v-
hasPerm="'erp:pi:supplier:link'" >
    <template #default="scope">
      <el-link type="success" :href="urlFormat(scope.row)" target="_blank" > <el-icon><Link

```


3.1 ?????????UserBuyer?

☐ ☐ t_user_buyer

```
@Data
@ApiModel(value="t_user_buyer☐☐", description="☐☐")
@EqualsAndHashCode(callSuper = true)
@TableName("t_user_buyer")
public class UserBuyer extends BaseEntity{
    @ApiModelProperty(value = "☐☐")
    @TableField(value = "name")
    private String name;

    @ApiModelProperty(value = "☐☐id")
    @TableField(value = "company")
    private String company;

    @ApiModelProperty(value = "☐☐")
    @TableField(value = "address")
    private String address;

    @ApiModelProperty(value = "☐☐☐")
    @TableField(value = "contact")
    private String contact;

    @ApiModelProperty(value = "☐☐☐")
    @TableField(value = "mobile")
    private String mobile;

    @ApiModelProperty(value = "token")
    @TableField(value = "token")
    private String token;

    @ApiModelProperty(value = "☐☐☐☐")
    @TableField(value = "tokentime")
    private Date tokentime;

    @ApiModelProperty(value = "☐☐☐☐")
    @TableField(value = "createtime")
```

```
private Date createtime;
}
```

3.2 ?????????UserSupplier?

☐ ☐ t_user_supplier

```
@Data
@ApiModel(value="UserSupplier☐☐", description="☐☐☐")
@EqualsAndHashCode(callSuper = true)
@TableName("t_user_supplier")
public class UserSupplier extends BaseEntity{
    @ApiModelProperty(value = "☐☐")
    @TableField(value = "name")
    private String name;

    @ApiModelProperty(value = "buyerid")
    @TableField(value = "buyerid")
    private String buyerid;

    @ApiModelProperty(value = "☐☐")
    @TableField(value = "address")
    private String address;

    @ApiModelProperty(value = "☐☐☐")
    @TableField(value = "contact")
    private String contact;

    @ApiModelProperty(value = "☐☐☐")
    @TableField(value = "mobile")
    private String mobile;

    @ApiModelProperty(value = "token")
    @TableField(value = "token")
    private String token;

    @ApiModelProperty(value = "password")
    @TableField(value = "password")
    private String password;
```

```

@ApiModelProperty(value = "令牌")
@TableField(value = "tokentime")
private Date tokentime;

@ApiModelProperty(value = "创建时间")
@TableField(value = "createtime")
private Date createtime;

@ApiModelProperty(value = "禁用")
@TableField(value = "disabled")
private Boolean disabled;
}

```

4. API????

4.1 ??API??

4.1.1 ??????API

类	方法	API	文件
OrderApi	orderApi.getBuyer({"token": state.token})	/quote/api/v1/quote/getBuyer	orderApi.js
OrderApi	orderApi.addBuyer(state.buyer)	/quote/api/v1/quote/addBuyer	orderApi.js
ThirdPartyApi	thirdpartyApi.getQuoteToken()	/erp/api/v1/thirdparty/getQuoteToken	thirdpartyApi.js
ThirdPartyApi	thirdpartyApi.saveQuoteToken(data)	/erp/api/v1/thirdparty/saveQuoteToken	thirdpartyApi.js
ThirdPartyApi	thirdpartyApi.unbindQuoteToken()	/erp/api/v1/thirdparty/unbindQuoteToken	thirdpartyApi.js

4.1.2 ??????API

类	方法	API	文件
SupplierApi	supplierApi.listsupplier(datas)	/quote/api/v1/quote/supplier/listSupplier	supplierApi.js
SupplierApi	supplierApi.deletesupplier({"id":id})	/quote/api/v1/quote/supplier/deleteSupplier	supplierApi.js
SupplierApi	supplierApi.addsupplier(token,data)	/quote/api/v1/quote/supplier/addSupplier/{token}	supplierApi.js

□□	□□□□	API□□	□□□□
□□□□	supplierApi.updateSupplier(data)	/quote/api/v1/quote/supplier/updateSupplier	supplierApi.js

4.2 ??API??

4.2.1 ???API??

```
// UserBuyerController.java (□□)
@RestController
@RequestMapping("/quote/api/v1/quote")
public class UserBuyerController {

    @Autowired
    private IUserBuyerService userBuyerService;

    @GetMapping("/getBuyer")
    public Result<UserBuyer> getBuyer(@RequestParam String token) {
        UserBuyer buyer = userBuyerService.getByToken(token);
        return Result.success(buyer);
    }

    @PostMapping("/addBuyer")
    public Result<String> addBuyer(@RequestBody UserBuyer buyer) {
        String token = userBuyerService.saveOrUpdateBuyer(buyer);
        return Result.success(token);
    }
}
```

4.2.2 ???API??

```
// ThirdPartyController.java (□□)
@RestController
@RequestMapping("/erp/api/v1/thirdparty")
public class ThirdPartyController {

    @GetMapping("/getQuoteToken")
    public Result<Map<String, Object>> getQuoteToken() {
        // □□□□□□□□□□□□□□□□
        Map<String, Object> tokenInfo = new HashMap<>();
    }
}
```

```

        // ...
        return Result.success(tokenInfo);
    }

    @PostMapping("/saveQuoteToken")
    public Result<String> saveQuoteToken(@RequestBody Map<String, Object> data) {
        //
        // ...
        return Result.success();
    }

    @GetMapping("/unbindQuoteToken")
    public Result<String> unbindQuoteToken() {
        //
        // ...
        return Result.success();
    }
}

```

4.2.3 API

```

// SupplierManagerController.java
@RestController
@RequestMapping("/quote/api/v1/quote/supplier")
public class SupplierManagerController {

    @Autowired
    private IUserSupplierService userSupplierService;

    @PostMapping("/listSupplier")
    public Result<List<UserSupplier>> listSupplier(@RequestBody Map<String, Object> data) {
        String token = (String) data.get("token");
        List<UserSupplier> suppliers = userSupplierService.listByBuyerToken(token);
        return Result.success(suppliers);
    }

    @DeleteMapping("/deleteSupplier")
    public Result<String> deleteSupplier(@RequestParam String id) {
        userSupplierService.removeById(id);
    }
}

```

```

        return Result.success();
    }

    @PostMapping("/addSupplier/{token}")
    public Result<String> addSupplier(@PathVariable String token, @RequestBody UserSupplier
supplier) {
        supplier.setBuyerid(token);
        userSupplierService.save(supplier);
        return Result.success();
    }

    @PutMapping("/updateSupplier")
    public Result<String> updateSupplier(@RequestBody UserSupplier supplier) {
        userSupplierService.updateById(supplier);
        return Result.success();
    }
}
}

```

5. ???????

5.1 ????????

1. `onMounted(() => loadToken())`
2. `loadToken()` `thirdpartyApi.getQuoteToken()`
3. `loadBuyer()` `loadSupplier()`
- 4.

5.2 ????????

- 1.
2. `bindToken()`
3. `bindToken()` `orderApi.getBuyer()`
4. `thirdpartyApi.saveQuoteToken()`
- 5.

5.3 ????????

5.3.1 ??????

1. 单击 "添加" 按钮时调用 `handleShow()` 方法
2. 单击 "添加" 按钮时调用 `CreateDialog` 方法
3. 单击 "添加" 按钮时调用 `handleShow()` 方法
4. 单击 "添加" 按钮时调用 `supplierApi.addsupplier()` 方法
5. 单击 "添加" 按钮时调用 `handleShow()` 方法

5.3.2 修改

1. 单击 "修改" 按钮时调用 `editSupplier()` 方法
2. 单击 "修改" 按钮时调用 `CreateDialog` 方法
3. 单击 "修改" 按钮时调用 `handleShow()` 方法
4. 单击 "修改" 按钮时调用 `supplierApi.updatesupplier()` 方法
5. 单击 "修改" 按钮时调用 `handleShow()` 方法

5.3.3 删除

1. 单击 "删除" 按钮时调用 `delSupplier()` 方法
2. 单击 "删除" 按钮时调用 `handleShow()` 方法
3. 单击 "删除" 按钮时调用 `supplierApi.deletesupplier()` 方法
4. 单击 "删除" 按钮时调用 `handleShow()` 方法

6. 数据库

6.1 数据库连接

1. 单击 "数据库" 按钮时调用 `handleShow()` 方法
2. 单击 "数据库" 按钮时调用 `handleShow()` 方法
3. 单击 "数据库" 按钮时调用 `handleShow()` 方法
4. 单击 "数据库" 按钮时调用 `handleShow()` 方法
5. 单击 "数据库" 按钮时调用 `handleShow()` 方法
6. 单击 "数据库" 按钮时调用 `handleShow()` 方法

6.2 数据库操作

1. 单击 "数据库" 按钮时调用 `handleShow()` 方法
2. 单击 "数据库" 按钮时调用 `handleShow()` 方法
3. 单击 "数据库" 按钮时调用 `handleShow()` 方法
4. 单击 "数据库" 按钮时调用 `handleShow()` 方法
 - 单击 "数据库" 按钮时调用 `handleShow()` 方法
 - 单击 "数据库" 按钮时调用 `handleShow()` 方法
 - 单击 "数据库" 按钮时调用 `handleShow()` 方法
 - 单击 "数据库" 按钮时调用 `handleShow()` 方法
5. 单击 "数据库" 按钮时调用 `handleShow()` 方法

- `URLSearchParams`

`URL`

1. `URLSearchParams`
2. `URLSearchParams`
3. `URLSearchParams`

8. ???????

8.1 ???????

8.1.1 ???????

`reactive` `URLSearchParams`

`toRefs` `URLSearchParams`

`ref`

```
// 初始化
const state = reactive({
  edittoken:"",
  editname:"",
  supplier:{},
  tableData:[],
  token:"",
  tokenname:"",
  title:'编辑',
  isowner:false,
  buyer:{'edit':true},
})
const{ token,buyer,edittoken,editname,supplier,tableData,title,isowner,tokenname
}=toRefs(state);

// 初始化
const edittoken = ref("");
const editname = ref("");
const supplier = ref({});
const tableData = ref([]);
const token = ref("");
const tokenname = ref("");
const title = ref('编辑');
```



```

    // 删除
}

// 用户买家信息
@Data
public class UserBuyer {
    @NotBlank(message = "姓名不能为空")
    private String name;

    @NotBlank(message = "手机号不能为空")
    @Pattern(regexp = "^1[3-9]\\d{9}$", message = "手机号格式不正确")
    private String mobile;

    // ...
}

```

8.2.2 删除

删除

```

@Service
public class UserSupplierServiceImpl implements IUserSupplierService {

    @Autowired
    private UserSupplierMapper userSupplierMapper;

    @Autowired
    private QuoteOrderSupplierMapper quoteOrderSupplierMapper;

    @Override
    @Transactional(rollbackFor = Exception.class)
    public boolean removeSupplier(String id) {
        // 删除
        boolean result = userSupplierMapper.deleteById(id) > 0;

        // 删除关联的报价单
        quoteOrderSupplierMapper.deleteBySupplierId(id);

        return result;
    }
}

```

