

一、使用SQL快速开发

1.新建动态表格模板代码

如: umppPlanOrderResolve_dy.vue

```
<template slot="umppPlanOrderResolve_dy">
  <div slot="center">
    <eh-tabs :tabs.sync="tabs" v-model="tabsName" ref="myTabs">
      <template slot="umppPlanOrderResolve-Slot">
        <eh-dyna-big-data-grid
          id="umppPlanOrderResolveDyGrid"
          ref="umppPlanOrderResolveDyGrid"
          @selection-change="handleSelectionChange"
          :options="{ isAdd: false, isEdit: false, isDel: false }"
          :dataSourceParams="{ }"
        >
        </eh-dyna-big-data-grid>
      </template>
    </eh-tabs>
  </div>
</template>

<script>
import qs from "qs";
export default {
  name: "umppPlanOrderResolve_dy",
  data() {
    return {
      rowData: [],
      activeNames: ["1"],
      tabs: [
        // 默认展示的所有标签页
        { name: "umppPlanOrderResolve-Slot", title: "订单分解" },
      ],
      tabsName: "umppPlanOrderResolve-Slot", // 默认展示并聚焦的标签页, key自定义
    };
  },
  methods: {
    //点击某一行获取数据
    handleSelectionChange(val) {
      this.rowData = val;
    },
  },
}
</script>

<style scoped>

</style>
```



如: poScheduleDataCheck.vue



```
<template slot="poScheduleDataCheck">
  <eh-layout>
    <div slot="center">
      <eh-tabs :tabs.sync="tabs" v-model="tabsName" ref="myTabs">
        <template slot="invalidProduBom-Slot">
          <eh-dyna-big-data-grid
            id="invalidProduBomDyGrid"
            ref="invalidProduBomDyGrid"
            @selection-change="handleSelectionChange"
            :options="{ isAdd: false, isEdit: false, isDel: false }"
            :dataSourceParams=""{}"
          >
        </eh-dyna-big-data-grid>
        </template>
        <template slot="invalidProduceCapacity-Slot">
          <eh-dyna-big-data-grid
            id="invalidProduceCapacityDyGrid"
            ref="invalidProduceCapacityDyGrid"
            @selection-change="handleSelectionChange"
            :options="{ isAdd: false, isEdit: false, isDel: false }"
            :dataSourceParams=""{}"
          >
        </eh-dyna-big-data-grid>
        </template>
        <template slot="invalidSapworkCenter-Slot">
          <eh-dyna-big-data-grid
            id="invalidSapworkCenterDyGrid"
            ref="invalidSapworkCenterDyGrid"
            @selection-change="handleSelectionChange"
            :options="{ isAdd: false, isEdit: false, isDel: false }"
            :dataSourceParams=""{}"
          >
        </eh-dyna-big-data-grid>
        </template>
      </eh-tabs>
    </div>
  </eh-layout>
</template>

<script>
import qs from "qs";
export default {
  name: "poScheduleDataCheck",
  data() {
    return {
      activeNames: ["1", "2", "3"],
      tabs: [
        // 默认展示的所有标签页
        { name: "invalidProduBom-Slot", title: "BOM结构检查" },
        { name: "invalidProduceCapacity-Slot", title: "产线能力配置检查" },
        { name: "invalidSapworkCenter-Slot", title: "产线成本中心检查" },
      ],
      tabsName: "invalidProduBom-Slot", // 默认展示并聚焦的标签页, key自定义
      rowData: [],
    };
  }
};
```

```

    },
    methods: {
      //点击某一行获取数据
      handleSelectionChange(val) {
        this.rowData = val;
      },
    }
  }
</script>

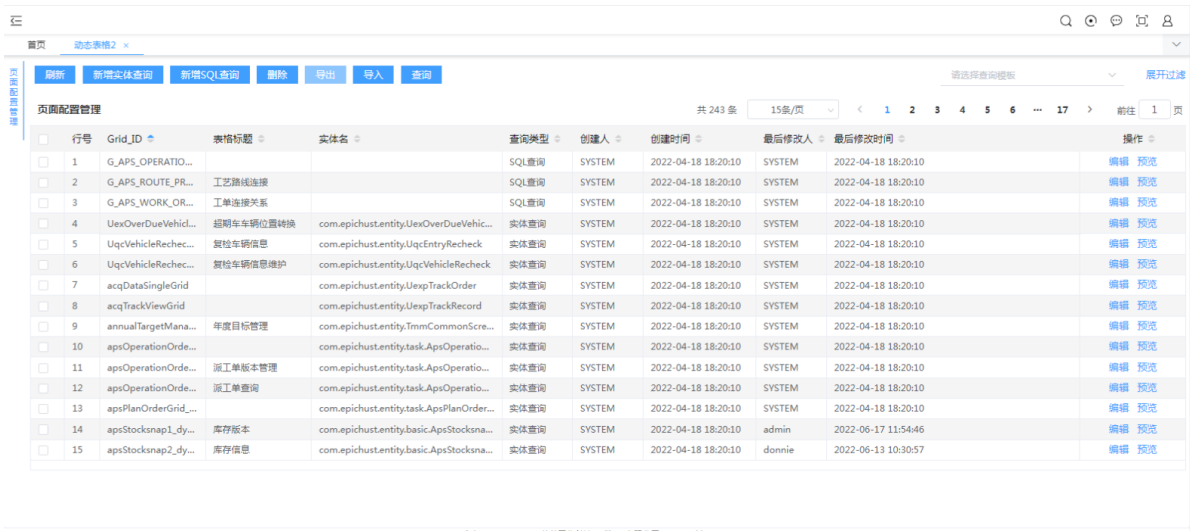
<style scoped>

</style>

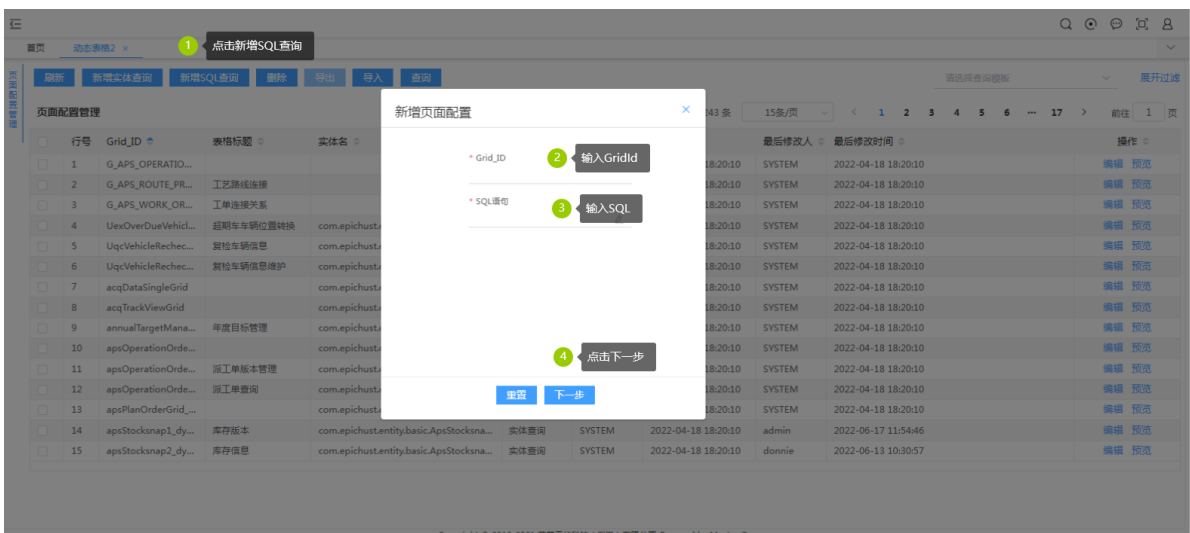
```

2.自定义GridId和SQL脚本

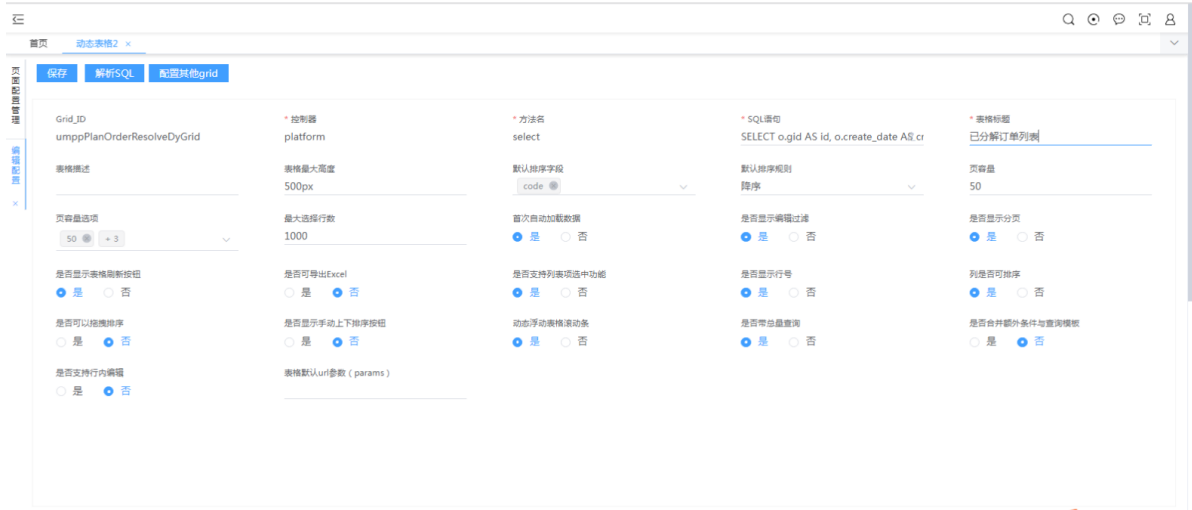
打开动态表格配置页面



点击新增SQL查询，配置GridId和SQL语句



设置标题



SQL脚本举例:

```

SELECT o.gid AS id, o.create_date AS createDate, o.create_id AS createId,
o.data_role AS dataRole, o.data_role1 AS dataRole1
, o.data_role2 AS dataRole2, o.deleted AS deleted, o.is_active AS isActive,
o.is_delete AS isDelete, o.modify_date AS modifyDate
, o.modify_id AS modifyId, o.uda1 AS uda1, o.uda1c AS uda1c, o.uda1n AS uda1n,
o.uda2 AS uda2
, o.uda2c AS uda2c, o.uda2n AS uda2n, o.uda3 AS uda3, o.uda3c AS uda3c, o.uda3n
AS uda3n
, o.uda4 AS uda4, o.uda4c AS uda4c, o.uda4n AS uda4n, o.uda5 AS uda5, o.uda5c
AS uda5c
, o.uda5n AS uda5n, o.assign_date AS assignDate, o.bom_code AS bomCode,
o.bom_state AS bomState, o.bom_type AS bomType
, o.bom_version AS bomVersion, o.code AS code, o.decompose_state AS
decomposeState, o.demand_flag AS demandFlag, o.drawing_version AS drawingVersion
, o.due_date AS dueDate, o.first_end_time AS firstEndTime, o.group_sap AS
groupsap, o.kitting_check_state AS kittingCheckState, o.lock_flag AS lockFlag
, o.mark_time AS markTime, o.measure AS measure, o.nature AS nature,
o.offline_date AS offlineDate, o.offline_seq AS offlineSeq
, o.offline_shift AS offlineShift, o.offline_shift_name AS offlineShiftName,
o.online_date AS onlineDate, o.online_seq AS onlineSeq, o.online_shift AS
onlineShift
, o.online_shift_name AS onlineShiftName, o.parent_id AS parentId,
o.planned_begin_date AS plannedBeginDate, o.planned_end_date AS plannedEndDate,
o.priority AS priority
, o.pro_type AS proType, o.process_line_gid AS processLineGid, o.produ_code AS
produCode, o.produ_name AS produName, o.produ_type AS produType
, o.product_unit AS productUnit, o.qty AS qty, o.remark AS remark,
o.res_fixed_flag AS resFixedFlag, o.route_gid AS routeGid
, o.route_version AS routeVersion, o.sales_order_code AS salesOrderCode,
o.sap_order_type AS sapOrderType, o.schedule_state AS scheduleState,
o.scheduled_begin_date AS scheduledBeginDate
, o.scheduled_end_date AS scheduledEndDate, o.seq AS seq, o.seq_fixed_flag AS
seqFixedFlag, o.site_code AS siteCode, o.source_type AS sourceType
, o.split_id AS splitId, o.split_qty AS splitQty, o.state AS state,
o.store_code AS storeCode, o.time_fixed_flag AS timeFixedFlag
, o.to_schedule AS toSchedule, o.type AS type, o.work_center_gid AS
workCenterGid,c.code as workCenterCode,c.name as workCenterName,l.code as
routeCode,l.name as routeName
FROM umpp_plan_order o
left join pmbf_work_center c on o.work_center_gid=c.gid and c.is_delete = 0

```

```
left join mbf_route_line l on o.route_gid = l.gid and l.is_delete = 0
WHERE o.is_delete = 0
and o.parent_id is null and o.decompose_state = 1
```

3.配置菜单

保存	重置				
选择父级菜单 菜单根节点 / 生产调度 / 订单管理	* 菜单名称 撤销分解 (动态)	关联URI fps/planorder/umpplOrderResolve_	* 维护url umpplOrderResolve_dy	图形报表URL 请选择	系统标识 WEB
打开方式 主选项卡	是否刷新 不刷新	是否展开 不展开	是否 不设置		
菜单标识 请选择	菜单来源 code	菜单样式	扩展字段		

4.调整个性化配置主要是按钮、查询模板和列属性

调整好属性后如下:

行号	订单号	订单类型	产品物料编码	产品物料名称	产品物料类型	数量	单位	拆分前数量	发布状态	排程状态	分解状态	下发状态	优先级	订单性质	工厂编码
1	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
2	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
3	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
4	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
5	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
6	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
7	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
8	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
9	PO-20220705-1278...	气缸套	10031107	气缸套6114东康54...	成品	400	PC		未发布	未排程	已分解	未下发		正常	
10	PO-20220705-1277...	气缸套	10019785	气缸套6130东康49...	成品	310	PC		未发布	未排程	已分解	未下发		正常	
11	PO-20220705-1277...	气缸套	10019785	气缸套6130东康49...	成品	310	PC		未发布	未排程	已分解	未下发		正常	
12	PO-20220705-1277...	气缸套	10019785	气缸套6130东康49...	成品	310	PC		未发布	未排程	已分解	未下发		正常	
13	PO-20220705-1277...	气缸套	10019785	气缸套6130东康49...	成品	310	PC		未发布	未排程	已分解	未下发		正常	

到此动态表格使用SQL快速开发完成

二、下面介绍常用的配置

1.自定义按钮

代码放在标签内

```
<template v-slot:grid-toolbar>
  <eh-button class="changeButton" @click="orderStructure"
:disabled="undodisab">{{
    $t("base.i18n_viewTheOrderStructure")
  }}</eh-button>
  <eh-button class="changeButton" v-throttle @click="undoSplitOrder"
:disabled="undodisab">{{
    $t("base.i18n_undoDecomposition")
  }}</eh-button>
</template>
```



2.注意POST请求参数不支持encodeURIComponent

```
id: encodeURIComponent(ids.toString())
//替代写法
id: ids.join(",")
```



3.配置过滤模板的选择器

静态配置：如

```
{
  "keyName": "value",
  "valueName": "key",
  "selectAttributes": {
    "clearable": true
  },
  "selectOptions": [
    {
      "key": "0",
      "value": "气缸套"
    },
    {
      "key": "1",
      "value": "机加"
    },
    {
      "key": "2",
      "value": "浇注"
    },
    {
      "key": "3",
      "value": "熔炼"
    }
  ]
}
```



高级配置

```
{
  "keyName": "value",
  "valueName": "key",
  "selectAttributes": {
    "clearable": true
  },
  "selectRequest": "bppComboxControllerEX!createResCombox.m?
hasHead=1&format=0&resType=S"
}
```



4.列字段类型

```
string
BigDecimal
date
```



5.时间格式化YYYY-MM-DD

代码放在标签内

```
<!-- 需求时间 -->
<template slot="render-dueDate" slot-scope="{ row }">
  {{ row.dueDate | dateFormat }}
</template>
```



6.表格行加颜色

```
{
  //行颜色
  "row-style": function({
    row,
    rowIndex
  }) {
    let style = {};
    let str = row.syncResult;
    let planQty = row.planQty;
    if (str == "success") {
      style = {
        backgroundColor: "#E0F8E0"
      };
    }
    if (str !== "success" && str !== "" && str !== null) {
      style = {
        backgroundColor: "#F3E2A9"
      };
    }
    if (planQty === "0") {
      style = {
        backgroundColor: "#F6CECE"
      };
    }
    return style;
  },
};
```



7.表格单元格加颜色

```
{
  //列字体颜色
  "cell-style": function({
    row,
    column,
    rowIndex,
    columnIndex
  }) {
    let style = {};
    let issueStr = row.sapOrderType;
    let col = column.property;

    if (col == "sapOrderType" && (issueStr === "0" || issueStr === "")) {
      style = {
        color: "red"
      };
    }
    return style;
  }
};
```



```

    },
  });

  --灰色背景
  ({
    //列字体颜色
    "cell-style": function({
      row,
      column,
      rowIndex,
      columnIndex
    }) {
      let style = {};
      if (column.title == "锁定标识" && row.lockFlag === 20) {
        style = {
          background: 'lightgray',
          color: "green"
        };
      } else if (column.title == "SAP下发" && row.sapOrderType === "1") {
        style = {
          background: 'lightgray',
          color: "green"
        };
      } else if (column.title == "SAP下发") {
        style = {
          color: "red"
        };
      }
      return style;
    },
  });

```

8.可编辑单元格


9.单元格跳转实现

单元格重新渲染

```

<template v-slot:render-code="{ row }">
  <el-link @click.native="goView(row)" type="primary" :underline="false">
    {{ row.code }}
  </el-link>
</template>

```



加入跳转方法

```

goView(row) {
  vue.$router.push({name: "umppPlanOrderView", params: row});
},

```



效果如下

已分解订单列表

共 91 条 50条/页 < 1 2 > 前往 1 页

行号	订单号	订单类型	产品物料编码	产品物料名称	产品物料类型	数量	单位	发布状态	下发状态	优先级	订单性质	工厂编码	工作中心名称	工艺路线
1	PO-20220705-1277_C009	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
2	PO-20220705-1277_C008	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
3	PO-20220705-1277_C007	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
4	PO-20220705-1277_C006	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
5	PO-20220705-1277_C005	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
6	PO-20220705-1277_C004	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
7	PO-20220705-1277_C003	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
8	PO-20220705-1277_C002	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
9	PO-20220705-1277_C001	气缸套	10019785	气缸套-6130-东康49...	成品	310	PC	未发布	未下发		正常			
10	PO-20220705-1276_C003	气缸套	10019773	气缸套-6116-东康54...	成品	505	PC	未发布	未下发		正常			
11	PO-20220705-1276_C002	气缸套	10019773	气缸套-6116-东康54...	成品	505	PC	未发布	未下发		正常			
12	PO-20220705-1276_C001	气缸套	10019773	气缸套-6116-东康54...	成品	505	PC	未发布	未下发		正常			
13	PO-20220705-1275_C033	气缸套	10019764	气缸套-6114-东康53...	成品	514	PC	未发布	未下发		正常			

点击跳转效果如下:

PO-20220705-1277_C009 订单详情

2022-07-05 19:37 C未发布

工作中心: | 产品物料: 10019785 | 气缸套-6130-东康4999962-免东康标识 | 成品 | 工艺路线: |

BOM信息: 10019785_V1 | 系统BOM | 未创建

顺序号:	优先级:	数量: 310	仓库编码:
分解状态: 已分解	排程状态: 未排程	生产方式: 自制	需求时间: 2022-06-30
订单性质: 正常	产品项目编号: PO-20220705-1277	图纸版本号:	最早开工时间:
最早开工时间:	销售订单编码:	创建人: donnie	创建时间: 2022-07-05 19:37
修改人: SYSTEM	修改时间: 2022-07-05 19:37	激活状态: 激活	备注:

10.以菜单的方式打开Tab页

跳转方法

```

goview(row) {
  vue.$router.push({name: "umppPlanOrderView", params: row});
},

```



注: 不希望单独配置 (显示) 菜单, 则需要在路由-index.js加入如下配置: 这样不会显示该菜单

```

{
  path: "/umppPlanOrderView",
  name: "umppPlanOrderView",
  title: t("base.i18n_orderDetailsInquiry"),
  component: () => import("@/views/umaplan/view/umppPlanOrderView"),
  meta: {
    name: "umppPlanOrderView",
    title: t("base.i18n_orderDetailsInquiry"),
    show: 0,
    requiresAuth: true,
    keepAlive: true,
    isTab: true,
  },
},

```



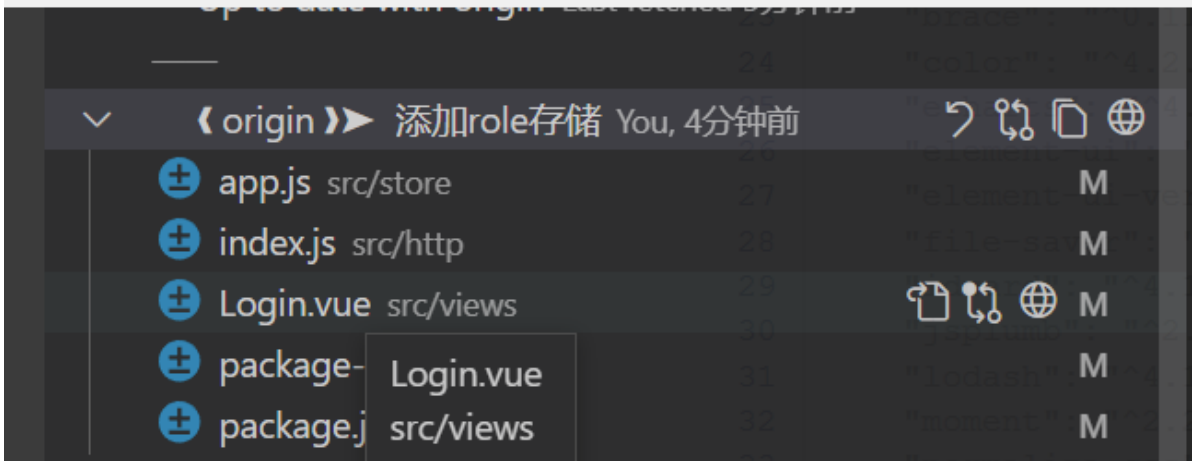
11.表格上编辑配置 (铅笔) 功能

目前有BUG, 只能修改dynamicGrid.component.js的源码

```
created() {  
  // this.isDynaGridEdit =  
  util.getCookie("role").includes("ROLE_DynaGrid_EDIT");  
  this.isDynaGridEdit = true;  
},
```



```
8 },  
9 // 组件创建后调用方法  
0 created() {  
1   this.isDynaGridEdit = storage.getItem("role") includes("ROLE_DynaGrid_EDIT");  
2   // this.$http.get("./dynamic-test.json").then((res) => {  
3     // console.log(res);  
4     // });  
5 },
```



动态表格快捷编辑不存在是没有获取本地存储role,已经添加role存储,代码已经提交, 依赖升级到了0.9.38@全体成员

12.自定义按钮功能

- 自定义按钮配置, 添加自定义按钮, 按钮会置于在表格上方区域
- 通过编辑函数方法, 实现按钮业务逻辑操作
- 注意事项:
 - 函数自带三个默认参数, 可以直接使用
selectedRows: 当前表格选择的数据
this: 当前页面对象
ehGrid: 当前表格对象
 - 函数中qs.stringify() 无法使用, 改用 this.\$util.getFormDataByObject()

行号	按钮ID	按钮名称	函数方法	是否隐藏
1	dyna_btn_default_add	新增	dynaGrid.add()	<input type="checkbox"/>
2	dyna_btn_default_edit	编辑	dynaGrid.edit()	<input type="checkbox"/>
3	dyna_btn_default_del	删除	dynaGrid.del()	<input type="checkbox"/>
4	dyna_btn_7572	锁定计划	if (selectedRows.length < 0) { this.\$message({ message: this.\$t("base.i18n_firstSelectTheRowOfDataToBeManipulated") + "!",	<input type="checkbox"/>
5	dyna_btn_9766	取消锁定计划	if (selectedRows.length < 0) { this.\$message({ message: this.\$t("base.i18n_firstSelectTheRowOfDataToBeManipulated") + "!",	<input type="checkbox"/>

- 按钮中的代码示例:

```
if (selectedRows.length < 0) {  
  this.$message({
```



```
message: this.$t("base.i18n_firstSelectTheRowOfDataToBeManipulated") +
"! ",
  type: "warning",
});
return;
}
let ids = [];
selectedRows.forEach((row) => {
  ids.push(row.id);
});
let url = "umppPlanOrderBatchUpdateController!setLockFlag.m";
this.$confirm('此操作将锁定选中的订单计划，是否继续?', this.$t('base.tips'), {
  confirmButtonText: this.$t('base.i18n_ok'),
  cancelButtonText: this.$t('base.i18n_cancel'),
  type: 'warning'
}).then(() => {
  this.$http.post(url, this.$util.getFormDataByObject({
    ids: ids,
    lockValue: 20
  })).then((res) => {
    if (res.code !== 500) {
      this.$message({
        message: res.data + "! ",
        type: res.type,
      });
      if (res.type === "success") {
        this.$refs.umppPlanOrderUpdate_dy.bindGrid();
      }
    }
  });
}).catch(() => {
  this.$message({
    type: "info",
    message: vue.$t("des.i18n_des_deletionCanceled"),
  });
});
});
```