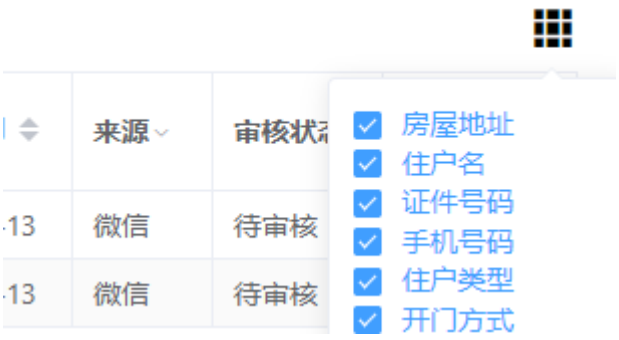
封装带分页、列显示/隐藏表格（配合导出）

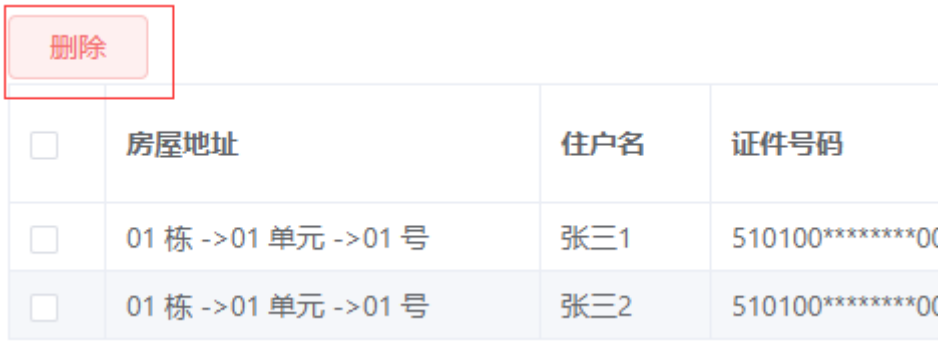
CustomTable组件

1. 组件封装了列显示/隐藏



2. 提供头部插槽

- **插槽名header**



3. props属性

- **查询url queryUrl**
  - 必传
  - 字符串
  - 查询对应的url

- **删除url deleteUrl**
  - 非必传

- 字符串
- 删除对应的url

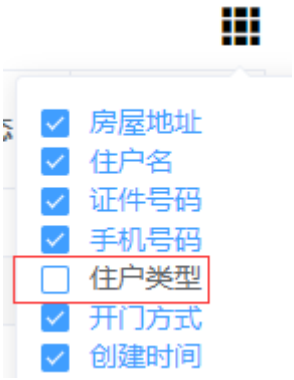
- **列配置 columns**
  - 必传
  - 数组
  - 列配置属性继承el-table Table-column Attributes的属性，拓展如下：

1. render：渲染函数
2. type：提供默认标题（label属性）
   - sex:封装性别默认渲染

   - certNo：证件号码默认实现

   - phone：电话号码
3. dict：字典默认翻译实现
4. filter：为true时**配合dict属性**自动生成数据过滤

5. isShow：列显示/隐藏初始配置。默认值为true，显示列；为false时隐藏列



- **操作列配置 operates**
  - 非必传
  - 对象
  - 列配置属性继承Table-column Attributes的属性，拓展如下：
    1. list: 操作功能配置（图标形式）

a. content: 图标提示文字

b. auth: 操作权限

c. method: 执行的方法，例如this.view

删除功能指定为'defaultDelete'即可，默认提供实现

d. class: 图标类必须两个类

名：'icon类'、'table-operation'

最后一个操作图标时多'no-margin-right'

- 插槽名operates，自定义操作列，封装当前行数据在row属性

- **字典 dict**
  - 非必传
  - 但是有要翻译的字典列时必传
  - 将父页面字典传入即可

- **表格的控制参数 options**
  - 非必传
  - 对象
  - 配置属性继承el-table Table Attributes的属性

4. 事件

- **selection-change**
  - 与el-table事件用法一致
  - **删除已封装，该事件可以不用**
- **row-dblclick**
  - 与el-table事件用法一致

表格使用示例

参考src/views/ai-estate/property/household/register/Index.vue

- ## 删除按钮绑定封装表格的batchDelete方法

封装表格默认提供了删除实现

关键代码：@click.native="$refs.CustomTable.batchDelete"

```
<el-dropdown-item v-has-permission="['household:delete']" @click.native="$refs.CustomTable.batchDelete">{{ $t('table.delete') }}</el-dropdown-item>
```

- ## 使用封装表格

```
<custom-table
 ref="CustomTable"
 :query-url="queryUrl"
 :delete-url="deleteUrl"
 :columns="columns"
 :operates="operates"
 :dict="dict"
 @search="search"
 @row-dblclick="onRowDblClick"
>
 <!-- 列插槽，名称为列prop属性 -->
 <template #authEndtime="{row}">
 <span v-if="row.authType === '1'">{{ transDict('auth_type',row.authType) }}</span>
 <span v-else>{{ row.authEndtime }}</span>
 </template>
</custom-table>
```

- ## 引用封装表格

```
import CustomTable from '@/components/CustomTable'
```

- ## 注册封装表格

```
components: { CustomTable }
```

- ## data属性

设置字典属性、表格属性、查询条件属性

```
data() {
 return {
 dict: {}, // 保存字典
```

```
// -----表格子组件相关属性
queryUrl: 'aiestate/household/list', // 查询url
deleteUrl: 'aiestate/household/', // 删除url
columns: [
{
prop: 'buildingNo,unitNo,houseNo',
 label: this.$t('household.address'),
 render: (h, data) => {
 const row = data.row
 return (
 <span >{row.buildingNo} 栋 ->{row.unitNo} 单元 ->{row.houseNo} 号</span>
 )
 },
 showOverflowTooltip: true,
 minWidth: 200
 },
 {
 prop: 'householdSex',
 width: 62,
 type: 'sex',
 dict: 'sex',
 align: 'center',
 filter: true
 },
 {
 prop: 'certNo',
 type: 'certNo',
 showOverflowTooltip: true,
 minWidth: 150
 },
 {
 prop: 'householdPhone',
 type: 'phone',
 showOverflowTooltip: true,
 minWidth: 110
 },
 {
 prop: 'householdType',
 label: this.$t('household.householdType'),
 width: 90,
```

```
  dict: 'household_type',
  filter: true
  }
  ], // 列配置
  operates: {
  width: 97,
  list: [
  {
  content: '查看',
  auth: 'household:view',
  method: this.view,
  class: 'el-icon-view table-operation'
  },
  {
  content: '删除',
  auth: 'household:delete',
  method: 'defaultDelete',
  class: 'el-icon-delete table-operation no-margin-right'
  }
  ]
  }, // 操作列配置
  queryParams: {}, // 查询参数
```

- **组件mount生命周期查询表格数据**

```
mounted() {
 this.search()
```

- **常见方法示例**

search：封装表格调用查询方法

reset：封装表格调用重置方法

onRowDblClick：封装表格双击行调用方法

```
methods: {
 search() {
 this.$refs.CustomTable.fetch({
 ...this.queryParams
 })
 }, // 搜索
```

```
reset() {
this.queryParams = {}
this.$refs.CustomTable.$refs.table.clearSort()
 this.$refs.CustomTable.$refs.table.clearFilter()
 this.search()
 }, // 重置
 onRowDblClick(row) {
 this.view(row)
 }, // 双击查看
```

导出

1. 前端

CustomTable组件基础上使用

- **引用方法**

```
import { exportExcel } from '@/utils/excelExport'
```

- **使用方法**

```
methods: {
 exportExcel, // 导出
```

2. 后端

参考cc/mrbird/febs/server/aiestate/controller/HouseholdController.java

- 先查询数量判断数量是否超过允许，然后查询选择列并导出

```
@PostMapping("export")
@PreAuthorize("hasAuthority('household:export')")
@ControllerEndpoint(operation = "导出住户", exceptionMessage = "导出住户失败")
public void export(HttpServletResponse response, Household household, String showCols) {
 int count = this.householdService.queryExportCount(household);
 //检查导出数据条数是否不超过允许最大导出条数
 if(!ExcelUtil.checkExportSize(response, count, 500)) {
 return;
 }
 //showCols 以逗号分隔的选中列的字段名
```

```
11    String[] split = showCols.split(",");
12    //查询数据
13    List<Household> list = this.householdService.export(household,split);
14    //导出数据
15    ExcelUtil.writeExcel(response, list, split, Household.class, "住户导出",
"住户导出");
16  }
```

- service关键代码

  只返回选择列

```
1  public List<Household> export(Household household, String[] fields) {
2    LambdaQueryWrapper<Household> queryWrapper = new QueryWrapper<Household>
().
3    select(camelToUnderscore(fields)).lambda();
```