TUI使用文档 V0.20.2

使用准备

基于jQuery、bootstrap、react、fontawesome(新版的tui.css已经包含该字体库样式)、simditor(可选)、jquery-form(可选)

- jQuery：//cdn.bootcss.com/jquery/1.11.3/jquery.min.js
- bootstrap
    - css：//cdn.bootcss.com/bootstrap/3.3.5/css/bootstrap.min.css
    - js：//cdn.bootcss.com/bootstrap/3.3.5/js/bootstrap.min.js
- font awesome：//cdn.bootcss.com/font-awesome/4.5.0/css/font-awesome.min.css
- react
    - react：//cdn.bootcss.com/react/0.14.6/react.min.js
    - react-dom：//cdn.bootcss.com/react/0.14.6/react-dom.min.js
- simditor：http://www.qintrend.net/resources/jsLib/trend/simditor/simditor.all.min.js
- jQuery-form：http://cdn.bootcss.com/jquery.form/3.51/jquery.form.min.js
- tui
    - css：
    - js：

表格－TUI.table 提供两个参数options和entity(可选)，说明如下

初始化表格
TUI.table(options, entity);

表格刷新
options.refresh = true;
TUI.table({options: options, entity: entity})

options 表格选项

属性	说明
type string	表格类型：table(标准表格)/tableTree(树形表格) table(标准表格)： tableTree(树形表格)：对返回值有要求，父对象中子级属性为children
container string	容器ID
formId string	表单ID，可选，不填会生成默认ID
contentType string	请求内容编码类型，默认值：application/x-www-form-urlencoded

submitBefore function	列表请求之前执行的方法
	事例：
	var demoTableOptions = {
	container: 'demoDOM',
	submitBefore: functionn(params) {
	// 目前有data一个参数，请求的数据
	console.log(params.data);
	},
	url: ''
	};

method string	远程请求方式 GET/POST，默认GET方式请求
url string	远程调用方法 返回值：对象，必须包含rows和rowCount属性，分别表示数据集合和总记录数

columns array	定义列属性
	返回值：数组，每列为一个对象
	事例：
	columns: [
	{
	// 列名称
	text: 'ID',
	// 列对应后台的字段名 支持多级调用
	field: 'id',
	// 该方法可对该列值进行特殊操作
	// 接受一个参数，为当前对象
	handle: function(obj) {
	},
	// 是否开启checkbox选项，默认false不开启
	// 此选项一般用在主键列上，并配合toolbar使用
	checkbox: true
	}
	]

rowHandles object	行操作，目前支持删除、查看和自定义操作
	返回值：对象
	事例：
	rowHandles: {
	// 删除，接收一个参数，为当前记录对象
	delete: function(obj) {
	return {
	method: 'get',
	url: url,
	// 回调函数，如果删除成功请返回 success
	callback: function(result) {
	return 'success';
	}
	};
	},
	// 查看，接收一个参数，为当前记录对象
	find: function(obj) {
	return {
	// 默认get方式
	method: 'get',
	url: '/teacher/get/' + obj.objectId,
	// 可选，回调函数
	// 如果你需要对得到的对象进行操作可以使用callback
	callback: function(result) {
	return result;
	}
	};
	},
	// 自定义行操作
	custom: [
	{
	// 操作名称
	text: '自定义操作',
	// 图标class 可选
	iconClass: 'fa fa-cog',
	// 处理方法，接受当前row对象
	handle: function(row) {
	alert('当前id是' + row.objectId);
	}
	}
	]
	}

searchbar array	筛选条件工具栏
	返回值：数组
	事例：
	searchbar: [
	{
	// 类型，默认text
	// 目前支持的类型: input select radio date
	type: 'select',
	// 筛选条件名称
	text: '项目类型',
	// 对应的后台字段
	field: 'companyOrPersonal',
	// 当type为select或radio时，需要用options填充值
	// select 支持远程加载
	options: [
	{ text: '不限' },
	{ text: '赣核盈', value: 'company' },
	{ text: '赣核贷', value: 'personal' }
	]
	}
	]
	
	当type为select时，options有三种赋值方法
	1.本地加载，也就是事例中所写的
	2.远程加载，options值为远程url，返回集合，集合内的对象要有text / value属性
	3.远程加载2，options值为对象，有三个属性：url，textField，valueField，
	这种方法更灵活，可以自定义text / value对应的属性名

toolbar array	功能性工具栏 目前支持button和checkbox
	返回值：数组
	事例：
	toolbar: [
	// button用法
	{
	// 按钮名称
	text: '删除',
	// class
	className: 'btn btn-danger’,
	// 处理方法
	handle: function() {
	}
	},
	// checkbox用法
	{
	text: ‘点我’,
	type: ‘checkbox’,
	id: checkbox id,
	name: checkbox name,
	// 是否选中 默认false
	checked: true,
	}
	]

pagination boolean	是否分页，默认true分页
maxSize int	每页最大显示记录数，默认10条

entity 实体类选项

属性	说明
key string	主键字段名
ajaxSubmit boolean	是否异步提交表单，目前依赖jquery.form.js

create object	创建实体
	事例：
	create: {
	// 方式一：该方式有三个参数，会走默认的创建实体表单
	method: 'POST',
	url: '/teacher/create',
	// 接收一个参数，返回结果，如果创建成功请返回success
	callback: function(result) {
	return 'success';
	}
	
	// 方式二：该方式为自定义表单按钮事件，不会触发默认的事件
	handle: function() {
	
	}
	}

update object	更新实体
	事例：
	update: {
	method: 'POST',
	url: '/teacher/update',
	// 接收一个参数，返回结果，如果创建成功请返回success
	callback: function(result) {
	return 'success';
	},
	// 可选，可以自定义是否显示修改按钮，返回success显示
	condition: funtion(entityData) {
	return 'success';
	}
	}

属性	说明

fields array	实体字段，添加或查看记录时显示的字段
	事例：
	fields: [
	{
	// 字段名称
	text: '性别',
	// 对应的后台字段名
	field: 'gender',
	// 类型 默认text 目前支持的字段类型: input select radio texture editor date
	type: 'select',
	// type为select或radio时需要使用options填充值
	options: [
	{ text: '男', value: 'MALE' },
	{ text: '女', value: 'FEMALE' }
	],
	// 验证器
	validators: {
	// 非空验证 message属性可不写
	notEmpty: {
	message: 'XX不能为空'
	},
	// 远程验证，返回json格式
	// {valid: true} true通过 false不通过
	remote: {
	url: '',
	data: {
	key: 'gender',
	value: 'gender'
	},
	message: '性别填写不正确'
	}
	},
	readOnly: false,
	disabled: false
	}
	]
	
	当type为select时，options有三种赋值方法
	1.本地加载，也就是事例中所写的
	2.远程加载，options值为远程url，返回集合，集合内的对象要有text / value属性

属性	说明

custom array	自定义实体操作
	事例：
	custom: [
	{
	// 操作名称
	text: '自定义操作',
	// 图标class 可选
	className: 'fa fa-cog',
	// 处理方法，接受当前row对象
	handle: function(row) {
	alert('当前id是' + row.objectId);
	// 返回success则会返回并刷新列表
	return 'success';
	}
	}
	]

模态框－TUI.Modal

```
var modalProps = {
        id: 'id', // 模态框ID，可选
        type: 'lg', // 模态框大小，lg（大）、中、sm（小）
        title: '标题',
        content: '内容', // 模态框内容，可以输入html
        // 确定事件
        confirm: function() {
                // 返回success关闭模态框
                return 'success';
        }
};

TUI.Modal(modalProps);
```

弹出提示信息框

```
TUI.success('成功');
TUI.danger('失败');
```

```
TUI.warning('警告');
TUI.info('信息');
```

加载提示框－TUI.loading({})

显示

```
TUI.loading({ text: '正在加载中' });
```

关闭

```
TUI.loading({ className: 'hide' });
```

工具类－TUI.Utils.xxx

格式化日期－传入毫秒

```
TUI.Utils.dateFormat(date, 'yyyy-mm-dd hh:mm:ss');
TUI.Utils.dateFormat(date, 'yyyy-mm-dd');
```