

Overview

```
Table
|-- Data
    |-- Row
        |-- Cell
            |-- Item
```

- **select**: User click one row
- **check**: User click the **checkbox** of one row, or **Ctrl/Shift+Click** in **multi** mode.
- **cancel** Unselect the checked rows.

Properties

fields

```
"fields" : {
  type : Array,
  default : ()=>[]
}
```

Defind each column of the table by `Array{Object}`, the elements in Array like:

```
{
  title : "i18n:xxx",      // column title
  nowrap : false,          // keep items nowrap
  width : "20%",           // field measure hint
  display : "theName"      // How to display the row data in cell
  //.....
  // In editing mode
  type : "String",
  dict : "xxx",
  comType : "ti-xxx",
  comConf : {/*..*/},
  transformer : Function,
  serializer : Function,
  //.....
  ignoreNil : true,        // If value is nil, ignore it
  //.....
  focusBy : "focus",     // auto set sub-com focus status
```

```
widthBy : "width" // auto set sub-com width measure
}
```

field measure hint

There are few valid value below:

- `auto` : keep the primary size
- `stretch` : auto assign the remains space
- `20%` : auto count the fixed size
- `0.3` : same as `30%`
- `400` : render as `400px`
- `-400` : min-width render as `400px` and stretch if necessary.
- `-0.3` : same as `30%` and stretch if necessary

The default value of field is `stretch`

field display

The field `display` defined how to render the cell of table. You can declare the value in three modes below:

Type	Description
<code>String</code>	key in raw data, rendered by <code>ti-label</code>
<code>Object</code>	customized the display component
<code>Array{String Object}</code>	multi rendering

It will be formatted to Array like:

```
[{
  key : "theName",          // How to pick out the value from row data
  uniqueKey : "theName",   // String form for `key`
  defaultAs : undefined,   // The default value if nil pick out
  //-----
  // Translate value by special dictionary
  // @see Wn.Dict.get
  dict : "DictionaryName",
  //-----
  // Transformer
  // - `type` for the default transformer
  // - `transformer` for the customized one in higher priority
  type : "String",         // @see Ti.Types
  transformer : "toStr",   // @see Ti.Types.getFuncByType()
  transNil : false,        // force to call transformer even the value is nil
  //-----
  // Skip `nil` value(`null` or `undefined`). Default is `true`
  ignoreNil : true,
  //-----
  // Component
```

```

comType : "ti-label",
comConf : {
  "..." : {...},    // ... will extends all value to comConf
  "val"   : "${=value}", // value from row data by key
  "obj"   : "${=..}",   // value for whole row data
  "age"   : "${info.age}", // value from row data
  "href"  : "(value)?/a/to?id=${value}", // test the value before render
  "isCurrent" : "${=isCurrent}", // parent row is activated
  "isChecked" : "${=isChecked}", // parent row is checked
  "isHover"   : "${=isHover}",   // parent row is hover
  "isActive"  : "${=isActive}",  // parent row is activated
  "rowId"     : "${=rowId}"      // parent row ID
}
}]

```

The **key** present the way how to pick the value from row data. It can be **String** or **Array**:

- **String** : dynamic picking or static value
 - **a.b.c** : as key path to get the value
 - **'Hello'** : as static value to render directly
 - **text+>/a/link?nm=\${name} : +>** Link open in new tab
 - **'More' ->/a/link?id=\${id} : ->** Link open in same tab
- **Array** : as key set to pick out a new object

Note!! If **key** is falsy, the field will be ignored

explainDict

```

"explainDict" : {
  type : Function,
  default : _.identity
}

```

If **dict** key was declared in field display item, it will apply this callback to explain the display value. The function accepted two arguments (**value**, **dict**).

- **value** is the display item raw value picked by **key**
- **dict** is the display item **dict** setting.

The function should return the explained value for given value. For supporting async fetch, **async function** has been allowed.

Usually it would been use in **id->displayName** translation for some refer field in raw-data.

idBy

```
"idBy" : {  
  type : [String, Function],  
  default : "id"  
}
```

Indicate the way to get `id` from raw data. The value could be:

- `String`: Key in raw data to pick out the ID
- `Function`: Customized method (`itemData, index`) to pick out the ID from raw data.

The `id` of the raw data is the unique identification of each record. The value of the `id` should be `String`.

rawDataBy

```
"rawDataBy" : {  
  type : [Object, String, Function],  
  default : _.identity  
}
```

Each time eval the row data, it will be invoked to get the raw-data.

- `Object`: Object mapping
 - `String`: Get the value of row data
 - `Function`: transform function with one argument (`itemInData`)
-

iconBy

```
"iconBy" : {  
  type : [String, Function],  
  default : _.identity  
}
```

Get the icon data from the row data

- `String`: Get the value of row data
 - `Function`: transform function with one argument (`itemInData`)
-

indentBy

```
"indentBy" : {  
  type : [String, Function],  
  default : null  
}
```

Get the row indent from the row data

- **String** : Get the value of row data
- **Function** : transform function with one argument (**itemInData**)

Default **null** will cause the indent as **0**.

className

```
"className" : {  
  type : String,  
  default : null  
}
```

Customized class selector name in root element of the table component.

extendFunctionSet

```
"extendFunctionSet" : {  
  type : Object,  
  default : ()=>({})  
}
```

Extend function set for **transformer** in each field **display**

data

```
"data" : {  
  type : Array,  
  default : ()=>[]  
}
```

The input raw data to render by the component.

changedId

```
"changedId" : {  
  type : String,
```

```
    default : null
  }
```

Indicate which row has been change. If changed, it will show up special mark to highlight the status.

The value will be `_.isEqual` with the result of `idBy`.

In whole table, only one record can be marked as **changed**.

currentId

```
"currentId" : {
  type : String,
  default : null
}
```

Indicate the current row outside from the component.

It will auto update the private status `myCurrentId` and `myLastIndex`

checkedIds

```
"checkedIds" : {
  type : Array,
  default : ()=>[]
}
```

Indicate the checked rows outside from the components.

It will auto update the private status `myCheckedId`

multi

```
"multi" : {
  type : Boolean,
  default : false
}
```

Support multi rows selecting. If `false`, only one row can be checked whatever the `checkedId`.

If `multi` mode is enabled. It will support `Shift+Click` and `Ctrl+Click` to multi-select the rows.

It is effected only when `selectable` is enabled.

checkable

```
"checkable" : {  
  type : Boolean,  
  default : false  
}
```

Show the checkbox at the begin or each row. Default is **false**.

selectable

```
"selectable" : {  
  type : Boolean,  
  default : true  
}
```

Can select

openable

```
"openable" : {  
  type : Boolean,  
  default : true  
}
```

Can double-click one row to **open**

cancelable

```
{  
  type : Boolean,  
  default : true  
}
```

If **true**, user can cancel selected current row by click blank area.

hoverable

```
{
  type : Boolean,
  default : true
}
```

If `true`, show hover style when mouse hover the table-row.

puppetMode

```
{
  type : Boolean,
  default : false
}
```

If `true`, it will not update private status when user selected any row:

- `myCheckedIds`
- `myCurrentId`
- `myLastIndex`

Only parent component can update them by relative properites:

- `checkedIds`
 - `currentId`
-

border

```
"border" : {
  type : String,
  default : "row",
  validator : (v)=>/^(row|column|cell|none)$/.test(v)
}
```

Indicate the border aspect.

- `none`: No border.
 - `row`: border for each row
 - `column`: border for each column
 - `cell`: border for each cell
-

autoScrollIntoView


```
"autoScrollIntoView" : {
  type : Boolean,
  default : true
}
```

Auto scroll the first highlight item into view

scrollIndex

```
"scrollIndex" : {
  type : Boolean,
  default : false
}
```

If select row by index, auto-scroll the index into range.

blankAs

```
"blankAs" : {
  type : Object,
  default : ()=>({
    icon : "zmdi-alert-circle-o",
    text : "empty-data"
  })
}
```

data

```
data: ()=>({
  myLastIndex: 0,      // The last row index selected by user
  myCurrentId: null,   // Current row ID
  myCheckedIds: {},    // Which row has been checked
  theData: [],         // The data evaluated into fix-form
  myViewportWidth: 0,  // Update-in-time, root element width
  myColSizes: {
    primary: [],       // Primary Max Col-Sizes
    amended: []        // The col-size to display in DOM
  }
})
```

Events

selected

When row selected/checked or canceled, it will emit the event with payload:

```
currentId : ID,      // current row ID
checkedIds : {        // All checked row id set
  [ID] : true
},
selected : [..],      // checked rows raw-data
current : {..}        // current row raw-data
```

item:changed

When cell display item **changed**, it will emit the event with payload:

```
name : {DiplayItemKey}, // the display item key
value : Any              // The display UI changed event payload
rowId : ID,              // Relative row data ID
data : {..}              // Relative row raw-data
```