UtilWidget Class Documentation

To use the feature of Util class you should follow these code pattern:

http

to use `http()` request functionally:

```
new UtilWidget().http(msg).then(function(d){
  //your code here
});
```

Or you can use Async/await approach

```
var responseData = await new UtilWidget().http(req);
```

state

`state()` functionality pop **success** or **error** message and also check if the message have valid tag or not.

```
new UtilWidget().state(msg, true).then(function(d){
  //set true for pop message default only check tag but pop nothing
  //default flag is false
  //your code here
});
```

Or you can use Async/await approach

```
var responseData = await new UtilWidget().state(req, true);
```

state and http together

To use `http()` and `state()` function together

```
new UtilWidget().http(msg)
.then(d=> new UtilWidget().state(d, true) )
.then(function(d){
  //your code here
});
```

Or you can use Async/await approach

```
var data = await new UtilWidget().http(req);
var responseData = await new UtilWidget().state(data, true);
```

regularGrid

`regularGrid()` build regular sortable plain table grid.

```
//initiate grid
var regularGrid = new UtilWidget().regularGrid(id, configCol);
```

```
//to load data
regularGrid.setRowData(data);

//to get All rows value
var allRows = regularGrid.getAllRows();
```

deletableGrid

`deletableGrid()` build table with `"build"` feature.

```
//initial grid
var delGrid = new UtilWidget().deletableGrid(id, configCol);
```

```
//load data into grid
delGrid.setRowData(data);

/*----to fit the grid----*/
delGrid.sizeColumnsToFit();

delGrid.build(function(d){
  return {
    delete:{
      msg: {
        type:"",
        data:{}
      }/*--- the delete msg here ----*/ ,
      key: '' /*--- name of the Field which will build ----*/
    }
    edit:{
      id:'',
      callback: function(){
        console.log(delGrid.edit);
        //your code here
      }
    }
  };
});

//to get All rows value
var allRows = delGrid.getAllRows();
```

editableGrid

`editableGrid()` build table with `"delete" & "edit"` feature.

```
var editGrid = new UtilWidget().editableGrid(id, configCol);
```

```
editGrid.setRowData(data);
/*----to fit the grid----*/
editGrid.sizeColumnsToFit();

editGrid.build(function(d){
  return {
    delete:{
      msg: {
        type:"",
        data:{}
      }/*--- the delete msg here ----*/ ,
      key: '' /*--- name of the Field which will build ----*/
    },
    edit:{
      msg: {
        type:"",
        data:{}
      }/*--- the delete msg here ----*/ ,
      id: '' ,/*--- name of the id which will build ----*/
      tab:{ /*-- if you dont want to use modal --*/
```

```
        tabId: '',
        position: 3
      }
    }
  }
});
```

To get Data back from server side

```
//to get data
 editGrid.edit.then(function(d){
         // d is the return data
 });

//  or you may use
var editData = await editGrid.edit;


//to get All rows value
var allRows = editGrid.getAllRows();
```

detailGrid

`detailGrid()` build detail table with `"delete", "edit" & "detail"` feature.

```
//to initiate grid
var detailGrid = new UtilWidget().detailGrid(id, configCol);
```

```
detailGrid.setRowData(data);
/*----to fit the grid----*/
detailGrid.sizeColumnsToFit();


detailGrid.build(function(d){
  return {
    delete:{
      msg: {
        type:"",
        data:{}
      }/*--- the delete msg here ----*/ ,
      key: '' /*--- name of the Field which will build ----*/
    },
    edit:{
      msg: {
        type:"",
        data:{}
      }/*--- the delete msg here ----*/ ,
      id: '', /*--- name of the id which will build ----*/

      tab:{ /*-- if you dont want to use modal --*/
        tabId: '',
        position: 3
      }
    },
    detail:{
      msg: {
        type:"",
        data:{}
      }/*--- the delete msg here ----*/ ,
      id: '',/*--- name of the id which will build ----*/
      tab:{ /*-- if you dont want to use modal --*/
        tabId: '',
        position: 3
      }
    }
  }
});
```

To get Data back from server side

```
editGrid.edit.then(function(d){
         // d is the return data
 });
 //or you may use
var editData = await editGrid.edit;

editGrid.detail.then(function(d){
         // d is the return data
 });
 //or you may use
var detailData = await editGrid.detail;

//to get All rows value
var allRows = detailGrid.getAllRows();
```