

创建一个自定义切片器

使用 `GeneralSlicerData` 类或者 `TableSlicerData` 类来创建自定义切片器。

`GeneralSlicerData` 类用于处理二维的数据。该类提供了筛选数据和获取筛选结果的相关方法。

你可以给一个切片器上附加多个切片器，每一个切片器可以筛选一列数据

当筛选完成时，其他切片器将会从 `GeneralSlicerData` 类中收到一个通知。

切片器将会从 `GeneralSlicerData` 获取到筛选后的数据，并更新 UI。

和 `GeneralSlicerData` 一起工作的切片器遵循以下步骤：

1. 使用你提供的数据来创建 `GeneralSlicerData`
2. 创建一个自定义切片器并挂载到 `GeneralSlicerData` 上。
3. 从 `GeneralSlicerData` 中获取数据列数据并更新 UI。
4. 响应 UI 中的事件并调用 `GeneralSlicerData` 中的 `doFilter` 方法。
5. 从 `GeneralSlicerData` 获取数据并更新 UI

你可以使用 `getData` 方法从某一列获取数据。

你可以使用 `getExclusiveData` 方法来获取去重之后的数据。

你可以使用 `doFilter` 方法来给列设置筛选器。

你可以使用 `doUnfilter` 方法来去掉列的筛选器。

更多方法，你可以参照：`GeneralSlicerData` 类。

你可以使用 `attachListener` 方法监听 `GeneralSlicerData` 实例中的事件。

你可以使用 `detachListener` 方法来取消监听 `GeneralSlicerData` 中的事件。

下图展示了一个自定义切片器。

Name:
☐ Bob
☒ Betty
☒ Alice

Name	City	Birthday
Betty	Washington	1972/7/3
Alice	NewYork	1964/3/2

示例代码

以下代码创建了一个自定义筛选器。

JavaScript

```
<!DOCTYPE html>
<html>
<head>
  <title>Spread.Sheets Sample</title>
  <link href="./css/gc.spread.sheets.excel2013white.10.x.x.css"
rel="stylesheet" type="text/css" />
  <script src="http://code.jquery.com/jquery-2.0.2.js"
type="text/javascript"></script>
  <script
src="./scripts/gc.spread.sheets.all.10.x.x.min.js"></script>
  <script type="text/javascript">
    //Define data source.
```

```

columnNames = ["Name", "City", "Birthday"];
data = [
    ["Bob", "NewYork", "1968/6/8"],
    ["Betty", "Washington", "1972/7/3"],
    ["Alice", "NewYork", "1964/3/2"]];
//Define custom slicer.
function MySlicer(container) {
    this.container = container;
    this.slicerData = null;
    this.columnName = null;
}
MySlicer.prototype.setData = function (slicerData, columnName)
{
    this.slicerData = slicerData;
    this.columnName = columnName;
    this.slicerData.attachListener(this);
    this.onDataLoaded();
}
MySlicer.prototype.onDataLoaded = function () {
    //create slicer dom tree.
    var columnName = this.columnName,
        exclusiveData =
this.slicerData.getExclusiveData(columnName);
    $(this.container).append($('' + this.columnName +
':</span>' + '<br />')));
    var domString = "";
    for (var i = 0; i < exclusiveData.length; i++) {
        domString += '<input type="checkbox" name="' +
columnName + '" value="' + exclusiveData[i] + '">';
        domString += '<span>' + exclusiveData[i] + '</span>';
        domString += '<br />';
    }
    $(this.container).append($(domString));
    //attach events to dom.
    var self = this;
    $("[name='" + self.columnName + "']").change(function () {
        var slicer = self,
            exclusiveData =
slicer.slicerData.getExclusiveData(slicer.columnName),
            parent = $(this).parent(),
            items = parent.children(),
            indexes = [];
        for (var i = 0, length = items.length; i < length; i++)
        {
            if (items[i].checked) {
                var value = items[i].value;
                if (!isNaN(parseInt(value))) {
                    value = parseInt(value);
                }
                indexes.push(exclusiveData.indexOf(value))
            }
        }
        if (indexes.length === 0) {
            slicer.slicerData.doUnfilter(slicer.columnName);
        } else {
            slicer.slicerData.doFilter(slicer.columnName, {
exclusiveRowIndex: indexes });

```

```

        }
    });
};
MySlicer.prototype.onFiltered = function () {
    //The following is an example of showing the filtered
result.
    var slicerdata = this.slicerData;
    var filteredRowIndex = slicerdata.getFilteredRowIndex();
    var trs = $listTable.find("tr");
    for (var i = 0; i < slicerdata.data.length; i++) {
        if (filteredRowIndex.indexOf(i) !== -1) {
            $(trs[i + 1]).show();
        } else {
            $(trs[i + 1]).hide();
        }
    }
}
//Define the show filtered result method.
function initFilteredResultList() {
    var tableStr = "<table border='1' cellpadding='0'
cellspacing='0'><tr>";
    for (var i = 0; i < columnNames.length; i++) {
        tableStr += "<th>" + columnNames[i] + "</th>";
    }
    tableStr += "</tr>";
    for (var i = 0; i < data.length; i++) {
        tableStr += "<tr>";
        for (var j = 0; j < data[i].length; j++) {
            tableStr += "<td>" + data[i][j] + "</td>";
        }
        tableStr += "</tr>";
    }
    tableStr += "</table>";
    $listTable = $(tableStr);
    $("#list").append($listTable);
}
$(document).ready(function () {
    //init filtered result list.
    initFilteredResultList();
    //create a custom slicer and add this slicer to the
"slicerContainer" div.
    var slicer = new MySlicer($("#slicerContainer")[0]);
    var slicerData = new
GC.Spread.Slicers.GeneralSlicerData(data, columnNames);
    slicer.setData(slicerData, "Name");
});
</script>
</head>
<body>
    <div id="slicerContainer" style="border:1px solid
gray;width:190px"></div>
    <hr/>

```

```

        <div id="list" style="width:300px;float:left"></div>
    </body>
</html>

```

以下代码创建了一个使用 `TableSlicerData` 类作为数据的筛选器。

HTML

```

<!DOCTYPE html>
<html>
<head>
    <title>Spread.Sheets Development Sample</title>
    <link href="./css/gc.spread.sheets.excel2013white.10.x.x.css"
rel="stylesheet" type="text/css" />
    <script src="http://code.jquery.com/jquery-2.0.2.js"
type="text/javascript"></script>
    <script
src="./scripts/gc.spread.sheets.all.10.x.x.min.js"></script>
    <script type="text/javascript">
        //Define data source.
        data = [
            { "Name": "Bob", "City": "NewYork", "Birthday": "1968/6/8"
},
            { "Name": "Bob", "City": "NewYork", "Birthday": "1968/6/8"
},
            { "Name": "Bob", "City": "NewYork", "Birthday": "1968/6/8"
},
            { "Name": "Bob", "City": "NewYork", "Birthday": "1968/6/8"
},
            { "Name": "Betty", "City": "Washington", "Birthday":
"1972/7/3" },
            { "Name": "Betty", "City": "Washington", "Birthday":
"1972/7/3" },
            { "Name": "Betty", "City": "Washington", "Birthday":
"1972/7/3" },
            { "Name": "Alice", "City": "NewYork", "Birthday":
"1964/3/2" },
            { "Name": "Alice", "City": "NewYork", "Birthday":
"1964/3/2" },
            { "Name": "Alice", "City": "NewYork", "Birthday":
"1964/3/2" }];
        //Define custom slicer.
        function MySlicer(container) {
            this.container = container;
            this.slicerData = null;
            this.columnName = null;
        }
        MySlicer.prototype.setData = function (slicerData, columnName)
    {
        this.slicerData = slicerData;
        this.columnName = columnName;
        this.slicerData.attachListener(this);
        this.onDataLoaded();
    }
    MySlicer.prototype.onDataLoaded = function () {
        //create slicer dom tree.
    }

```

```

        var columnName = this.columnName,
            exclusiveData =
this.slicerData.getExclusiveData(columnName);
        $(this.container).append($('<span>' + this.columnName +
':</span>' + '<br />'));
        var domString = "";
        for (var i = 0; i < exclusiveData.length; i++) {
            domString += '<input type="checkbox" name="' +
columnName + '" value="' + exclusiveData[i] + '">';
            domString += '<span>' + exclusiveData[i] + '</span>';
            domString += '<br />';
        }
        $(this.container).append($(domString));
        //attach events to dom.
        var self = this;
        $("[name='" + self.columnName + "']").change(function () {
            var slicer = self,
                exclusiveData =
slicer.slicerData.getExclusiveData(slicer.columnName),
                parent = $(this).parent(),
                items = parent.children(),
                indexes = [];
            for (var i = 0, length = items.length; i < length; i++)
            {
                if (items[i].checked) {
                    var value = items[i].value;
                    if (!isNaN(parseInt(value))) {
                        value = parseInt(value);
                    }
                    indexes.push(exclusiveData.indexOf(value))
                }
            }
            if (indexes.length === 0) {
                slicer.slicerData.doUnfilter(slicer.columnName);
            } else {
                slicer.slicerData.doFilter(slicer.columnName, {
exclusiveRowIndex: indexes });
            }
        });
    };
    MySlicer.prototype.onFiltered = function () {
        //Sync the status if the data has been filtered by the
Spread.Sheets table.
        var slicerdata = this.slicerData;
        var exclusiveIndexes =
slicerdata.getFilteredIndexes(this.columnName);
        $.each($("#slicerContainer").children("input"), function
(i,input) {
        });
    }
    $(document).ready(function () {
        var spread = new
GC.Spread.Sheets.Workbook(document.getElementById("ss"), {sheetCount:1});
        var sheet = spread.getActiveSheet();
        //create a custom slicer and add this slicer to the
"slicerContainer" div.
        var slicer = new MySlicer($("#slicerContainer")[0]);

```

```
        var table = sheet.tables.addFromDataSource("table1", 1, 1,
data);
        var slicerData = table.getSlicerData();
        slicer.setData(slicerData, "Name");
    });
</script>
</head>
<body>
    <div id="slicerContainer" style="border:1px solid
gray;width:190px"></div>
    <hr />
    <div id="ss" style="width:500px;height:500px;border:1px solid
```

```
gray"></div>  
</body>  
</html>
```