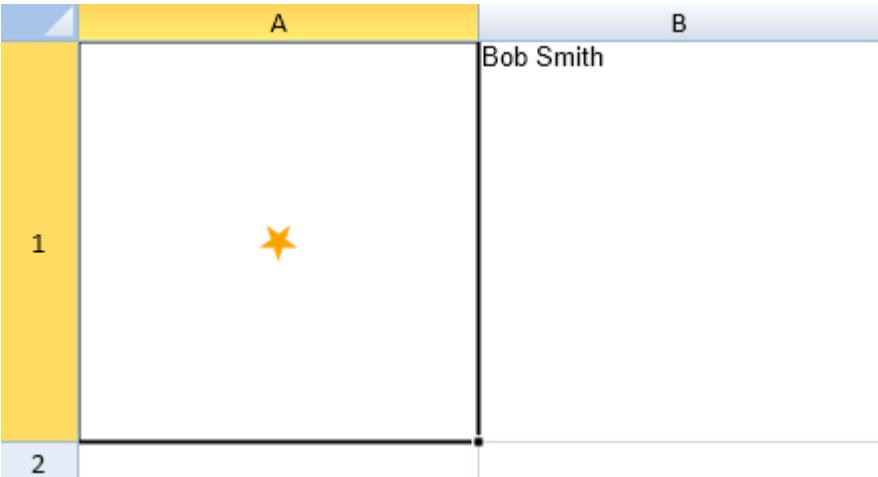


创建自定义单元格

你可以使用 `Base` 类来创建自己的单元格。



示例代码

以下代码创建了一个自定义单元格。

JavaScript

```
//Custom Cell Type
function FivePointedStarCellType() {
  this.fillStyle = "orange";
  this.size = 10;
}
FivePointedStarCellType.prototype = new
GC.Spread.Sheets.CellTypes.Base();
FivePointedStarCellType.prototype.paint = function (ctx, value,
x, y, w, h, style, context) {
  if (!ctx) {
    return;
  }
  ctx.save();
  // draw inside the cell's boundary
  ctx.rect(x, y, w, h);
  ctx.clip();
  ctx.beginPath();
  if (value) {
    ctx.fillStyle = this.fillStyle;
  } else {
    ctx.fillStyle = "gray";
  }
  var size = this.size;
  var dx = x + w / 2;
  var dy = y + h / 2;
  ctx.beginPath();
  var dig = Math.PI / 5 * 4;
  for (var i = 0; i < 5; i++) {
    ctx.lineTo(dx + Math.sin(i * dig) * size, dy +
Math.cos(i * dig) * size);
  }
  ctx.closePath();
  ctx.fill();
  ctx.restore();
}
```

```

    };
    FivePointedStarCellType.prototype.getHitInfo = function (x, y,
cellStyle, cellRect, context) {
        var xm = cellRect.x + cellRect.width / 2,
            ym = cellRect.y + cellRect.height / 2,
            size = 10;
        var info = { x: x, y: y, row: context.row, col:
context.col, cellRect: cellRect, sheetArea: context.sheetArea };
        if (xm - size <= x && x <= xm + size && ym - size <= y && y
<= ym + size) {
            info.isReservedLocation = true;
        }
        return info;
    };
    FivePointedStarCellType.prototype.processMouseUp = function
(hitInfo) {
        var sheet = hitInfo.sheet;
        if (sheet && hitInfo.isReservedLocation) {
            var row = hitInfo.row, col = hitInfo.col, sheetArea =
hitInfo.sheetArea;
            var newValue = !sheet.getValue(row, col, sheetArea);
            var cellEditInfo = { row: row, col: col, newValue:
newValue };
            spread.commandManager().execute({cmd: "editCell", sheetName:
activeSheet.name(), row: row, col: col, newValue: newValue});
            return true;
        }
        return false;
    };
    function FullNameCellType() {
    }
    FullNameCellType.prototype = new
GC.Spread.Sheets.CellTypes.Base();
    FullNameCellType.prototype.paint = function (ctx, value, x, y,
w, h, style, context) {
        if (value) {

GC.Spread.Sheets.CellTypes.Base.prototype.paint.apply(this, [ctx,
value.firstName + "." + value.lastName, x, y, w, h, style, context]);
        }
    };
    FullNameCellType.prototype.updateEditor = function
(editorContext, cellStyle, cellRect, context) {
        if (editorContext) {
            $(editorContext).width(cellRect.width);
            $(editorContext).height(100);
        }
    };
    FullNameCellType.prototype.createEditorElement = function
(context) {
        var div = document.createElement("div");
        var $div = $(div);
        $div.attr("gcUIElement", "gcEditingInput");
        $div.css("background-color", "white");
        $div.css("position", "absolute");
        $div.css("overflow", "hidden");
        $div.css("border", "2px #5292f7 solid");
    };

```

```

        var $span1 = $("<span></span>");
        $span1.css("display", "block");
        var $span2 = $("<span></span>");
        $span2.css("display", "block");
        var $input1 = $("<input type='text' />");
        var $input2 = $("<input type='text' />");
        $div.append($span1);
        $div.append($input1);
        $div.append($span2);
        $div.append($input2);
        return div;
    };
    FullNameCellType.prototype.getEditorValue = function
(editorContext) {
        if (editorContext && editorContext.children.length === 4) {
            var input1 = editorContext.children[1];
            var firstName = $(input1).val();
            var input2 = editorContext.children[3];
            var lastName = $(input2).val();
            return { firstName: firstName, lastName: lastName };
        }
    };
    FullNameCellType.prototype.setEditorValue = function
(editorContext, value) {
        if (editorContext && editorContext.children.length === 4) {
            var span1 = editorContext.children[0];
            $(span1).text("First Name:");
            var span2 = editorContext.children[2];
            $(span2).text("Last Name:");
            if (value) {
                var input1 = editorContext.children[1];
                $(input1).val(value.firstName);
                var input2 = editorContext.children[3];
                $(input2).val(value.lastName);
            }
        }
    };
    FullNameCellType.prototype.isReservedKey = function (e,
context) {
        //cell type handle tab key by itself
        return (e.keyCode === GC.Spread.Commands.Key.tab &&
!e.ctrlKey && !e.shiftKey && !e.altKey);
    };
    FullNameCellType.prototype.isEditingValueChanged = function
(oldValue, newValue, context) {
        if (newValue.firstName != oldValue.firstName ||
newValue.lastName != oldValue.lastName) {
            return true;
        }
        return false;
    };
    $(function () {
        var spread = new
GC.Spread.Sheets.Workbook(document.getElementById("ss"), {sheetCount:3});
        var sheet = spread.getActiveSheet();
        sheet.suspendPaint();
        sheet.setCellType(0, 0, new FivePointedStarCellType());
    });

```

```
sheet.setValue(0, 0, true);
sheet.setRowHeight(0, 200);
sheet.setColumnWidth(0, 200);
sheet.setCellType(0, 1, new FullNameCellType());
sheet.setColumnWidth(1, 200);
sheet.setValue(0, 1, { firstName: "Bob", lastName: "Smith"}
```

```
});  
        sheet.resumePaint();  
    });
```

另见

`setCellType` Method