

JS API

-
- ```

graph TD
 Root(()) --- HTML((HTML))
 Root --- JS((JS))
 HTML --- width((width))
 HTML --- height((height))
 JS --- left((left))
 JS --- Top((Top))
 left --- name((name))
 left --- publishUUUIUIDYellowfin 9.5((publishUUUIUIDYellowfin 9.5))
 Top --- widgetUUUIUIDYellowfin 9.5((widgetUUUIUIDYellowfin 9.5))
 Top --- onRender((onRender))
 onRender --- onRemove((onRemove))
 onRender --- API1[API]
 API1 --- JS2(JS)
 API1 --- CSS1(CSS)
 JS2 --- CSS2(CSS)
 JS2 --- API2[API]
 CSS1 --- API3[API]
 CSS1 --- API4[API]

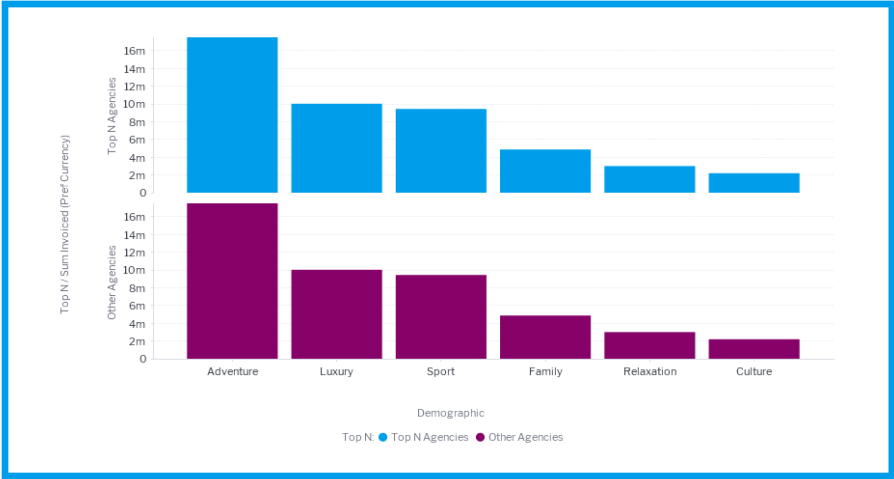
```



HTML

HTMLHTMLHTML

2



HTML

```
<canvas-area xmlns="http://www.w3.org/1999/xhtml" canvas-uuid="aee7c574-3ad3-4ac0-a366-7810861d91eb">
<!-- This the element tag for the Report-->
 <report-output publish-uuid="ffe09245-8739-416c-b039-05fff6427242" report-uuid="c83357db-8aef-4ec7-ab72-fce34de9ee77" height="475" top="46" left="54" name="Agency Benchmark" width="897" display-type="CHART" chart-uuid="a9aba052-51b5-47b1-b311-2c0cc3abd932" on-click="none" style="z-index: 3;"></report-output>
<!-- Element tag for the rectangle -->
 <canvas-rectangle publish-uuid="d0fafeb9-57a8-4a21-ab6c-844501ebf50b" width="927" height="496" top="35" left="36" rotation="0" name="Rectangle" stroke-color="#009eec" fill-color="#ffffff" on-click="none" style="z-index: 2" stroke-width="14" fill-opacity="100"></canvas-rectangle>
</canvas-area>
```

HTMLchart-uuiddisplay-typeREPORT

 Yellowfin 9.5UUIDUUIDUUIDYellowfinUUID

```
<report-output publish-uuid="ffe09245-8739-416c-b039-05fff6427242" report-uuid="c83357db-8aef-4ec7-ab72-fce34de9ee77" height="475" top="46" left="54" name="Agency Benchmark" width="897" display-type="REPORT" on-click="none" style="z-index: 3;"></report-output>
```

 Yellowfin 9.5UUIDUUIDUUIDYellowfinUUID

Top N	Company Name	Demographic	Sum Invoiced (Pref Currency)
Top N Agencies	Bargain Trips	Adventure	\$10,856
Top N Agencies	Bargain Trips	Family	\$295,827
Top N Agencies	Bargain Trips	Luxury	\$1,032,056
Top N Agencies	Bargain Trips	Relaxation	\$80,660
Top N Agencies	Bargain Trips	Sport	\$4,043
Top N Agencies	Beyond Adventure Holidays	Adventure	\$1,830,220
Top N Agencies	Beyond Adventure Holidays	Family	\$158,556
Top N Agencies	Big Foot Adventures	Adventure	\$2,933,095
Top N Agencies	Big Foot Adventures	Family	\$196,735
Top N Agencies	Buzz Adventures	Adventure	\$3,739,961
Top N Agencies	Buzz Adventures	Family	\$268,224
Top N Agencies	Coloured World Travel	Culture	\$265,321

width

height

left

Top

name

DOM

publishUUIDUUIDYellowfin 9.5

ID

widgetUUIDUUIDYellowfin 9.5

IDYellowfin 9UUID

JS

## JSJavaScript

```
// Declare variables or insert code here

/**
 * Called when the canvas and all its child elements are rendered and attached into the page.
 */
this.onRender = function () {
 // Insert your code here. This is an ideal place for setting up event listeners
};

/**
 * Called when the canvas and all its child elements are removed from the page.
 */
this.onRemove = function () {
 // Insert your code here. This is an ideal place for removing event listeners
};
```

### onRender

```
this.onRender = function() { //Your logic here }
```

### onRemove

```
let subTabAPI = this.apis.subtab;

this.bodyClick = function() {
 console.log('A click was recorded on ' + subTabAPI.name);
};

this.onRender = function() {
 document.body.addEventListener('click', this.bodyClick);
};

this.onRemove = function() {
 document.body.removeEventListener('click', this.bodyClick);
};
```

## API

this.apis

- 
- 
- 
-

```
this.apis.canvas;
this.apis.subtab;
this.apis.dashboard;
this.apis.filters
```

APIAPIthis.apis

API

APIAPIselectselectAllAPIselectselectAllquerySelectorquerySelectorAll

report\_1APIselectAPI

```
let reportElement = this.apis.canvas.select('report_1');
reportElement.onReportLoad.then(() => { //onReportLoad is a promise that is resolved once the reportAPI for that report element has finished loading.
 let reportAPI = reportElement.reportAPI; //This is now the ReportAPI.
});
```

*dashboard.loadReport()* *dashboard getAllReports()* *getReportsForSubTab()* [API](#)

JS

HTMLJSHTML

CSS

CSSCSSCSS

5

HTML

```
<canvas-area xmlns="http://www.w3.org/1999/xhtml" canvas-uuid="aee7c574-3ad3-4ac0-a366-7810861d91eb">
 <report-output publish-uuid="ffe09245-8739-416c-b039-05fff6427242" report-uuid="c83357db-8aef-4ec7-ab72-fce34de9ee77" height="475" top="46" left="54" name="report_1" width="897" display-type="CHART" chart-uuid="a9aba052-51b5-47b1-b311-2c0cc3abd932" on-click="none" style="z-index: 3;"></report-output>
 <canvas-rectangle publish-uuid="d0fafeb9-57a8-4a21-ab6c-844501ebf50b" width="927" height="496" top="35" left="36" rotation="0" name="Rectangle" stroke-width="14" stroke-color="#009eec" fill-opacity="100" fill-color="#ffffff" on-click="none" style="z-index: 2;"></canvas-rectangle>
</canvas-area>
```



Yellowfin 9.5UUIDUUUIDUUUIDYellowfinUUID

## JS

```
this.onRender = function () {
 //Select the report_1 element from the page canvas through the canvas api
 let report1 = this.apis.canvas.select('report_1');

 //The report will begin loading once it is added to the DOM, so it is possible that at this point it hasn't completed yet. So wait for the onReportLoad
 promise to resolve.
 report1.onReportLoad.then() => {
 let changeDisplayType = () => {
 let currentDisplay = report1.getAttribute('display-type');
 if(currentDisplay === "CHART") {
 currentDisplay = "REPORT";
 } else {
 currentDisplay = "CHART";
 }
 report1.setAttribute('display-type', currentDisplay);
 }
 this.reportChangeTimeout = setInterval(changeDisplayType, 5000);
 });
};

this.onRemove = function () {
 //The SubTab is being removed from the page, we want to stop triggering the reportChange
 clearTimeout(this.reportChangeTimeout);
};
```

## API

### HTML

```
<canvas-area xmlns="http://www.w3.org/1999/xhtml" canvas-uuid="a4c909fb-70d1-44f1-a95a-17a0eada6bf6">
 <custom-html widget-uuid="eff0b9bd-4009-4d83-ade8-46c40587ed97" width="880" height="450" top="130" left="10" rotation="0" name="custom
_table" on-click="none" style="z-index: 2;"></custom-html>
 <report-output widget-uuid="90a601cc-a148-40d8-a460-15daab8a238b" report-uuid="3e842fae-02f7-4ad3-a632-ca267e0078da" height="125" to
p="4" left="8" name="Campaign Summary" width="880" display-type="CANVAS" on-click="none" style="z-index: 1;"></report-output>
</canvas-area>
```

## JS

```

this.outputTypes = []; //Create an array of outputKeys and what report API they came from for easy removal

this.onRender = function () {
 let campaignSummary = this.apis.canvas.select('Campaign Summary');
 campaignSummary.onReportLoad.then(() => {
 let reportAPI = campaignSummary.reportAPI;
 let outputKey = reportAPI.registerOutputType('dataset', reportDataset => {
 let tableData = [];
 let tableElement = document.createElement('table');

 //Give the table a class name so we can style it
 tableElement.className = "codeModeExampleTable";

 let tHead = document.createElement('thead');
 let headerRow = document.createElement('tr');
 tHead.appendChild(headerRow);
 tableElement.appendChild(tHead);

 //Loop over all the reports fields to get their names. The fields will be
 //in the same order as the columns in the dataset
 reportAPI.fields.forEach(field => {
 let th = document.createElement('th');
 th.innerText = field.name;
 headerRow.appendChild(th);
 });

 let tbody = document.createElement('tbody');
 tableElement.appendChild(tbody);
 //Loop over the rows in the dataset and create a table row at each level.
 reportDataset.forEach(rowData => {
 let tr = document.createElement('tr');

 //Loop over the columns and append the formattedValue from each item
 rowData.forEach(columnData => {
 let td = document.createElement('td');
 //columnData will have rawValue, formattedValue and htmlFormattedValue.
 td.innerText = columnData.formattedValue;
 tr.appendChild(td);
 });
 tbody.appendChild(tr);
 });

 //Now we've done all that we want to get the custom_table element and insert the table there
 let customTable = this.apis.canvas.select('custom_table');
 customTable.innerHTML = ""; //Clear any previous table that might be there
 customTable.appendChild(tableElement); //Append our new table

 });

 this.outputTypes.push({
 key: outputKey,
 reportAPI: reportAPI
 });
 });
};

this.onRemove = function () {
 //Remove the output types so that we aren't generating extra tables each time we come back to this
 //tab
 this.outputTypes.forEach(outputObject => {
 outputObject.reportAPI.removeOutputType(outputObject.key)
 });
};

```

## CSS

```
table.codeModeExampleTable {
 width:100%;
 height:100%;
 border-collapse:separate;
 border-spacing:0px;
}

table.codeModeExampleTable thead {
 background-color: #009EEC;
}

table.codeModeExampleTable thead th {
 border-left: 1px solid white;
}

table.codeModeExampleTable tbody td {
 border:1px solid #ccc;
 border-top:none;
}
```