



Table of Contents

1.	Entry Point for JavaScript in atlas.html	1
2.	General Functions.....	1
2.1.	Map Navigation	1
2.2.	Map Features.....	2
2.3.	Map Feature Selection.....	3
2.4.	Filtering	3
2.5.	Map Layers	4
2.6.	Data.....	4
2.7.	Interface	6
3.	Template Specific Functions.....	7
3.1.	Single Map	7
3.2.	Double Map.....	8
3.3.	Double Map Time Series	9
3.4.	Bubble Plot.....	10
3.5.	Area Profile	11
4.	InstantAtlas Support.....	11



1. Entry Point for JavaScript in atlas.html

The file atlas.html contains the following JavaScript by default. The `init()` function is called by the atlas when it has completed loading. You should place your initialisation code in the `init()` function.

```
// Get the atlas object.
function getAtlas(appName)
{
    if (navigator.appName.indexOf ("Microsoft") != -1)
    {
        return window[appName];
    }
    else
    {
        return document[appName];
    }
}
var myAtlas = getAtlas("instantAtlasReport");

// Called by the atlas when it has completed loading.
function init()
{
    // Startup code goes here.
}
```



For most of the following functions add a two on the end of the function call to refer to the second data set e.g. `zoomIn2()` when using templates containing two maps

2. General Functions

2.1. Map Navigation

zoomIn():void

Zooms in on the map

zoomOut():void

Zooms out on the map

zoomFull():void

Zooms to the full extents of the map

zoomToBBox(bbox:String):void

Zooms to a feature in the map

@param **bbox**: A space separated list of the form "xmin ymin xmax ymax"

addMapBBoxListener(fnc:String):void

Listens for the event dispatched by the map when its bounding box has changed

@param **fnc**: The name of the function to call when the maps bounding box has changed



The function receives a bounding box which is a space separated list of the form "xmin ymin xmax ymax"

Example:

```
function init()
{
    myAtlas.addMapBBoxListener("bBoxListener");
}

function bBoxListener(bBox)
{
    alert(bBox);
}
```

2.2. Map Features

getMapFeature(id:String):Object

Returns information about the specified feature

@param id: The feature id

@return: An object containing the feature id and name and current value.

Example:

```
var o = myAtlas.getMapFeature("22015");
alert(o.id+" "+o.name+" "+o.value);
```

zoomToFeature(id:String):Boolean

Zooms to a feature in the map

@param id: The feature id

@return: true, if the zoom worked, otherwise false

addMapFeatureListener(fnc:String):void

Listens for feature events dispatched by the atlas

@param fnc: The name of the function to call when an event is dispatched

The function receives an event type and a list of hit features.

eventType: The type of event:

Event	Description
<i>change</i>	Indicates the selection in the map has changed
<i>featureClick</i>	Indicates the user clicked the mouse button over a map feature
<i>featureDoubleClick</i>	Indicates the user double clicked the mouse button over a map feature
<i>featureMouseDown</i>	Indicates the user pressed the mouse button over a map feature
<i>featureMouseUp</i>	Indicates the user released the mouse button over a map feature
<i>featureMouseMove</i>	Indicates the user moved the mouse pointer while hovering over a map feature
<i>featurerMouseOver</i>	Indicates the user rolled the mouse pointer over a map feature
<i>featureMouseOut</i>	Indicates the user rolled the mouse pointer out of a map feature

hitFeatures: A comma separated list of ids of the form "eh10,eh11,eh12"

Example:

```
function init()
{
    myAtlas.addMapFeatureListener("featureListener");
}

function featureListener(eventType, hitFeatures)
{
    if (eventType == "change")
    {
        alert("The following features were
            selected: "+hitFeatures);
    }
    else if (eventType == "featureClick")
    {
        alert("The following feature was clicked:
            "+hitFeatures);
    }
}
```

2.3. Map Feature Selection

selectFeature(id:String, on:Boolean):Boolean

Selects a feature

@param id: The feature id

@param on: Whether the feature should be selected or deselected

@return: true, if the feature was selected, otherwise false

selectFeatures(ids:String):void

Selects a list of features

@param ids: A comma separated list of feature ids of the form "eh10,eh11,eh12"

getSelectedFeatures():String

Returns a list of features in the current selection

@return: A comma separated list of feature ids of the form "eh10,eh11,eh12"

clearSelection():void

Clears the current selection

2.4. Filtering

filterFeatures(ids:String):void

Filters a list of features

@param ids: A comma separated list of feature ids of the form "eh10,eh11,eh12"

getFilteredFeatures():String

Returns a list of features in the current filter

@return: A comma separated list of feature ids of the form "eh10,eh11,eh12"

filterSelection():void

Filters the current selection

clearFilter():void

Clears the current filter

2.5. Map Layers

getMapLayer(id:String):Object

Returns information about the specified layer

@param id: The layer id

@return: An object containing the layer id and name.

Example:

```
var o = myAtlas.getMapLayer("_louisiana_counties.shp");
alert(o.id+" "+o.name);
```

getMapLayers():String

Gets the ids of the layers in the map

@return: A comma separated list of layer ids of the form "layer1,layer2,layer3"

Example:

```
var layerList = myAtlas.getMapLayers();
alert(layerList);
```

setMapLayerVisible(id:String, on:Boolean):void

Sets the visibility of a map layer

@param id: The layer id

@param on: true or false

2.6. Data

addDataListener(fnc:String):void

Listens for data events dispatched by the atlas when data change

@param fnc: The name of the function to call when an event is dispatched

The function receives an event type and an object containing the current geog id, theme id, indicator id and date.

eventType: The type of event:

Event	Description
<i>indicatorChanged</i>	Indicates an indicator changed
<i>themeChanged</i>	Indicates a theme changed
<i>geogChanged</i>	Indicates a geography changed



object: An object containing the current geog id, theme id, indicator id and date.

Example:

```
function init()
{
    myAtlas.addDataListener("dataListener");
}

function dataListener(eventType, o)
{
    alert(eventType+" "+o.geographyId+" "+o.themeId+"
        "+o.indicatorId+" "+o.date);
}
```

setData(geogId:String, themeId:String, indicatorId:String, date:String =>):void

Sets the current indicator

@param **geogId:** The geography id

@param **themeId:** The theme id

@param **indicatorId:** The indicator id

@param **date:** The date

getDataSummary():String

Returns the list of geographies

@return: A comma separated list of geography ids of the form "geog1,geog2"

getGeogSummary(id:String):Object

Returns information about the requested geography

@param **id:** The geography id

@return: An object containing the geog id, name and geometry, and a comma separated list of themes

Example:

```
var o = myAtlas.getGeogSummary("Louisiana_Parishes");
alert(o.id+" "+o.name+" "+o.geometry+" "+o.themes);
```

getThemeSummary(geogId:String, themeId:String):Object

Returns information about the requested theme

@param **geogId:** The geography id

@param **themeId:** The theme id

@return: An object containing the theme id and name, and a comma separated list of indicators

Example:

```
var o = myAtlas.getThemeSummary("Louisiana_Parishes",
    "t0");
alert(o.id+" "+o.name+" "+o.indicators);
```



getIndicatorSummary(geogId:String, indicatorId:String):Object

Returns information about the requested indicator

@param **geogId**: The geography id

@param **indicatorId**: The indicator id

@return: An object containing the indicator id and name, and a comma separated list of dates

Example:

```
var o = myAtlas.getIndicatorSummary("Louisiana_Parishes",
  "i0");
alert(o.id+" "+o.name+" "+o.dates);
```

getCustomProperty(id:String):String

Returns the value for the requested custom property

@param **id**: The property id

@return: The property value

openMetaData(id:String, target:String=" blank"):Boolean

Opens a metadata page for the requested feature

@param **id**: The feature id

@param **target**: The url target.

_blank for a new window

_self for the current page, top for the topmost frame container

_parent for the parent of the current frame container

@return: true if a url link exists, otherwise false

2.7. Interface**openShapeWin():void**

Opens the shape editor window

openTextWin():void

Opens the text editor window

openPrintWin():void

Opens the print preview window

openAboutWin():void

Opens the about window

openExportWin():void

Opens the export window

openNotesPage():void

Opens the notes page for the current indicator

resetLayout():void

Resets the atlas layout

toggleMap():void

Toggles the map visibility

toggleTable():void

Toggles the table visibility

toggleLegend():void

Toggles the legend visibility

toggleDataExplorer():void

Toggles the data explorer visibility

toggleFilterExplorer():void

Toggles the filter explorer visibility

toggleMetaData():void

Toggles the metadata box visibility

toggleSearch():void

Toggles the Google Maps search box visibility

3. Template Specific Functions

3.1. Single Map

toggleTimeAnimation():void

Toggles the time animation slider

toggleGeogExplorer():void

Toggles the geography explorer

toggleTimeSeries():void

Toggles the time series

toggleBarChart():void

Toggles the bar chart

togglePieChart():void

Toggles the pie chart

toggleAdvancedPieChart():void

Toggles the advanced pie chart

toggleComparisonTable():void

Toggles the comparison table

toggleFlippedBarChart():void

Toggles the flipped bar chart

toggleDiscreteTimeSeries():void

Toggles the discrete time series

toggleFlippedTimeSeries():void

Toggles the flipped time series

toggleBoxAndWhisker():void

Toggles the box and whisker chart

togglePyramidChart():void

Toggles the pyramid chart

toggleStats():void

Toggles the statistics box

toggleDotPlot():void

Toggles the dot plot

3.2. Double Map**toggleScatterPlot():void**

Toggles the scatter plot

toggleGeogExplorer():void

Toggles the geography explorer

toggleMap2():void

Toggles the map visibility for the second data set

toggleLegend2():void

Toggles the legend visibility for the second data set

toggleDataExplorer2():void

Toggles the data explorer visibility for the second data set

toggleMetaData2():void

Toggles the metadata box visibility for the second data set

3.3. Double Map Time Series

toggleTimeSeries():void

Toggles the time series

toggleBarChart():void

Toggles the bar chart

togglePieChart():void

Toggles the pie chart

toggleComparisonTable():void

Toggles the comparison table

toggleTimeSeries2():void

Toggles the time series for the second data set

toggleBarChart2():void

Toggles the bar chart for the second data set

togglePieChart2():void

Toggles the pie chart for the second data set

toggleComparisonTable2():void

Toggles the comparison table for the second data set

toggleMap2():void

Toggles the map visibility for the second data set

toggleTable2():void

Toggles the table visibility for the second data set

toggleLegend2():void

Toggles the legend visibility for the second data set

toggleDataExplorer2():void

Toggles the data explorer visibility for the second data set

toggleFilterExplorer2():void

Toggles the filter explorer visibility for the second data set

toggleMetaData2():void

Toggles the metadata box visibility for the second data set

3.4. Bubble Plot

toggleScatterPlot():void

Toggles the bubble plot

toggleGeogExplorer():void

Toggles the geography explorer

toggleTimeAnimation():void

Toggles the time animation slider

toggleGeogExplorer():void

Toggles the geography explorer

toggleTimeSeries():void

Toggles the time series

toggleBarChart():void

Toggles the bar chart

togglePieChart():void

Toggles the pie chart

toggleAdvancedPieChart():void

Toggles the advanced pie chart

toggleComparisonTable():void

Toggles the comparison table

toggleLegend2():void

Toggles the legend visibility for the second data set

toggleDataExplorer2():void

Toggles the data explorer visibility for the second data set

toggleDataExplorer3():void

Toggles the data explorer visibility for the third data set

toggleDataExplorer4():void

Toggles the data explorer visibility for the fourth data set

3.5. Area Profile

toggleSpineChart():void

Toggles the spine chart

toggleGeogExplorer():void

Toggles the geography explorer

toggleTimeSeries():void

Toggles the time series

toggleBarChart():void

Toggles the bar chart

togglePieChart():void

Toggles the pie chart

toggleAdvancedPieChart():void

Toggles the advanced pie chart

toggleComparisonTable():void

Toggles the comparison table

4. InstantAtlas Support

You can find all core user guides and a wide range of other InstantAtlas support resources at www.instantatlas.com/support.xhtml. InstantAtlas customers can login to their My InstantAtlas account and download a wide range of support resources and access a searchable knowledgebase. If these resources do not provide a solution, please contact your support provider.

