

# table.dataTable

## Description

Creates a <table> element using data retrieved via REST. The data returned from the REST call needs to be an array of objects

## Spawner Widget Name

```
kind: table.dataTable
```

## JavaScript Method

```
XNAT.table.dataTable([dataArray], {opts})
```

## Configuration Options

```

dataTable:
  kind: table.dataTable
  name: fooTable
  load: /data/url | $.path.to.data
  [data]: [
    { id: 1, firstname: 'Carl', lastname: 'Carlson', username: 'ccarlson', website: 'carl-carlson.
name' },
    { id: 2, firstname: 'Lenny', lastname: 'Leonard', username: 'lenny1', website: 'leonnyard.org'
},
    { id: 3, firstname: 'Bob', lastname: 'Robertson', username: 'brobertson', website: 'bobertson.
net' }
  ]
  [path]: "object.path.to.data"
  [header]: true
  [table]:
    id: 'user-list-table'
    className: 'xnat-table compact'
    style:
      width: 100%
  [items]:
    id: ~!
    _username: '~data.username' # adds a [data-username] attribute to each row
    firstname: First Name
    lastname: Last Name
    username: Username
    website:
      label: Website
      sort: true
      content: <a href="__VALUE__">Website</a>
    email:
      label: Email
      className: email
    filter: true
      apply: >
      function(email){
        return '<a href="mailto:" + email + "'>' + this.firstname + ' ' + this.lastname + '</a>'
      }
  [id]: foo-table
  [dataType]: json
  [sortable]: true -or- "list, of, sortable, columns"
  [filter]: "firstname, lastname, username, email"
  [container]: "#parent-element"
  [width]: 300px
  [element]:
    className: foo bar
    title: "This is your data, and it's rendering one row at a time."
  [before]: "Text, HTML, or Spawner object(s) to insert before the <table> element."
  [after]:
    smallTxt:
      tag: small.msg
      content: "This small message is brought to you by Spawner. It will be inserted after the <table>
element."

```

## Properties

Name	Value	Description
kind	panel. dataTable	Spawner widget type - creates a <table> element that loads data from a url or specific data set in <b>data</b> property.
name	(string)	Name attribute for this input, which must match the property name used for data loading and submission.
header	(boolean)	Should the property names of the first data object be used for the column headers? This does not affect output if there are <b>items</b> specified.

items	(object)	<p>Object map of properties to include in the rendered table. The property name here must match the property name in the data object, and the value will be used as the label for the header &lt;th&gt; element.</p> <p>Properties with a value of "<b>! </b>" will be hidden columns. The data will be in the table, but the associated &lt;th&gt; and &lt;td&gt; elements will be hidden.</p> <p>Properties with a value of "<b>-data</b>" will have their actual value added as <b>data-*</b> attributes in the parent &lt;tr&gt; element.</p> <p>If an object is used for an item's value, the <b>label</b> value will be used for the header, and the <b>content</b> value is HTML (with string replacement for '<b>__VALUE__</b>') that will be rendered in the table cells. The <b>call</b> property can be the name of an existing function (that must be defined before the table renders), or an anonymous function that can take a single argument of the cell's value and 'this' is the parent object so you can access other properties.</p>
id	(string)	Value for spawned element's id attribute.
load/url	(url)	REST URL to retrieve data
data	(array)	Explicitly pass data as an array of data objects or 2D array.
path	(string)	<p>Custom path to use for looking up object data returned from url specified in the <b>load</b> option (not supported with use of <b>data</b> property). This is a string that represents the object path in dot notation, or if there are property names that contain periods (WHY???), colons can be used as the path delimiter, and must prefix the path, like:</p> <pre>path: ":object:path:to:data.source"</pre>
dataType	json	What kind of data will be returned from the request? JSON is supported and is the default - XML support is planned.
sortable	(boolean / comma-separated list)	Are the columns sortable? Enter true to make ALL columns sortable, or a comma-separated list of the columns that can be sorted. The column names must match the object property names from the returned data.
filter	comma-separated list	List of property names (columns) that should be filterable. Adding items to the <b>filter</b> list will insert a row under the table header containing a text input where the filter text is entered.
container	(string - selector)	Selects existing DOM element that this table will be appended to (inserted into). If the <b>container</b> property is used, the table will not be rendered in the usual Spawner manner - it will not show up as a child of a parent element if it's in the parent's <b>contents</b> .
width	(string - CSS width)	Explicitly set CSS 'width' property. Will default to '100%' if not specified.
element	(object)	Object map of DOM attributes and properties to be directly attached to this table. Any attribute that can be added natively to a DOM element can be used in the <b>element</b> object.