DataTables example Generated content for a column

In some tables you might wish to have some content generated automatically. This can be done in a number of ways:

* [columns.render](http://datatables.net/reference/option/columns.render) for content that is dynamic (i.e. based upon the row's data)
* [columns.defaultContent](http://datatables.net/reference/option/columns.defaultContent) for static content (i.e. simple strings)

This examples shows the use of [columns.defaultContent](http://datatables.net/reference/option/columns.defaultContent) to create a *button* element in the last column of the table. A simple jQuery click event listener is used to watch for clicks on the row, and when activated uses the [row().data()](http://datatables.net/reference/api/row().data()) method to get the data for the row and show a bit of information about it in an alert box. This is a simple use case, but it can be built up to be arbitrarily complex.

Note also that the [columns.data](http://datatables.net/reference/option/columns.data) option for the column has been set to null to indicate that the column has no information that should be obtained data source object.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Position | Office | Extn. | Start date | Salary |
| Name | Position | Office | Extn. | Start date | Salary |

* Javascript
* HTML
* CSS
* Ajax
* Server-side script

The Javascript shown below is used to initialise the table shown in this example:

$(document).ready(function() { var table = $('#example').DataTable( { "ajax": "data/arrays.txt", "columnDefs": [ { "targets": -1, "data": null, "defaultContent": "<button>Click!</button>" } ] } ); $('#example tbody').on( 'click', 'button', function () { var data = table.row( $(this).parents('tr') ).data(); alert( data[0] +"'s salary is: "+ data[ 5 ] ); } ); } );

In addition to the above code, the following Javascript library files are loaded for use in this example:

* [//code.jquery.com/jquery-1.12.3.min.js](http://code.jquery.com/jquery-1.12.3.min.js)
* [../../media/js/jquery.dataTables.js](http://docs.google.com/media/js/jquery.dataTables.js)

The HTML shown below is the raw HTML table element, before it has been enhanced by DataTables:

This example uses a little bit of additional CSS beyond what is loaded from the library files (below), in order to correctly display the table. The additional CSS used is shown below:

The following CSS library files are loaded for use in this example to provide the styling of the table:

* [../../media/css/jquery.dataTables.css](http://docs.google.com/media/css/jquery.dataTables.css)

This table loads data by Ajax. The latest data that has been loaded is shown below. This data will update automatically as any additional data is loaded.

The script used to perform the server-side processing for this table is shown below. Please note that this is just an example script using PHP. Server-side processing scripts can be written in any language, using [the protocol described in the DataTables documentation](http://datatables.net/manual/server-side).

Other examples

[**Basic initialisation**](http://docs.google.com/basic_init/index.html)

* [Zero configuration](http://docs.google.com/basic_init/zero_configuration.html)
* [Feature enable / disable](http://docs.google.com/basic_init/filter_only.html)
* [Default ordering (sorting)](http://docs.google.com/basic_init/table_sorting.html)
* [Multi-column ordering](http://docs.google.com/basic_init/multi_col_sort.html)
* [Multiple tables](http://docs.google.com/basic_init/multiple_tables.html)
* [Hidden columns](http://docs.google.com/basic_init/hidden_columns.html)
* [Complex headers (rowspan and colspan)](http://docs.google.com/basic_init/complex_header.html)
* [DOM positioning](http://docs.google.com/basic_init/dom.html)
* [Flexible table width](http://docs.google.com/basic_init/flexible_width.html)
* [State saving](http://docs.google.com/basic_init/state_save.html)
* [Alternative pagination](http://docs.google.com/basic_init/alt_pagination.html)
* [Scroll - vertical](http://docs.google.com/basic_init/scroll_y.html)
* [Scroll - vertical, dynamic height](http://docs.google.com/basic_init/scroll_y_dynamic.html)
* [Scroll - horizontal](http://docs.google.com/basic_init/scroll_x.html)
* [Scroll - horizontal and vertical](http://docs.google.com/basic_init/scroll_xy.html)
* [Language - Comma decimal place](http://docs.google.com/basic_init/comma-decimal.html)
* [Language options](http://docs.google.com/basic_init/language.html)

[**Advanced initialisation**](http://docs.google.com/advanced_init/index.html)

* [DOM / jQuery events](http://docs.google.com/advanced_init/events_live.html)
* [DataTables events](http://docs.google.com/advanced_init/dt_events.html)
* [Column rendering](http://docs.google.com/advanced_init/column_render.html)
* [Page length options](http://docs.google.com/advanced_init/length_menu.html)
* [Multiple table control elements](http://docs.google.com/advanced_init/dom_multiple_elements.html)
* [Complex headers with column visibility](http://docs.google.com/advanced_init/complex_header.html)
* [Read HTML to data objects](http://docs.google.com/advanced_init/object_dom_read.html)
* [HTML5 data-\* attributes - table options](http://docs.google.com/advanced_init/html5-data-options.html)
* [HTML5 data-\* attributes - cell data](http://docs.google.com/advanced_init/html5-data-attributes.html)
* [Language file](http://docs.google.com/advanced_init/language_file.html)
* [Setting defaults](http://docs.google.com/advanced_init/defaults.html)
* [Row created callback](http://docs.google.com/advanced_init/row_callback.html)
* [Row grouping](http://docs.google.com/advanced_init/row_grouping.html)
* [Footer callback](http://docs.google.com/advanced_init/footer_callback.html)
* [Custom toolbar elements](http://docs.google.com/advanced_init/dom_toolbar.html)
* [Order direction sequence control](http://docs.google.com/advanced_init/sort_direction_control.html)

[**Styling**](http://docs.google.com/styling/index.html)

* [Base style](http://docs.google.com/styling/display.html)
* [Base style - no styling classes](http://docs.google.com/styling/no-classes.html)
* [Base style - cell borders](http://docs.google.com/styling/cell-border.html)
* [Base style - compact](http://docs.google.com/styling/compact.html)
* [Base style - hover](http://docs.google.com/styling/hover.html)
* [Base style - order-column](http://docs.google.com/styling/order-column.html)
* [Base style - row borders](http://docs.google.com/styling/row-border.html)
* [Base style - stripe](http://docs.google.com/styling/stripe.html)
* [Bootstrap 3](http://docs.google.com/styling/bootstrap.html)
* [Foundation](http://docs.google.com/styling/foundation.html)
* [jQuery UI ThemeRoller](http://docs.google.com/styling/jqueryUI.html)
* [Bootstrap 4 (Tech. preview)](http://docs.google.com/styling/bootstrap4.html)
* [Semantic UI (Tech. preview)](http://docs.google.com/styling/semanticui.html)
* [Material Design (Tech. preview)](http://docs.google.com/styling/material.html)
* [UIKit (Tech. preview)](http://docs.google.com/styling/uikit.html)

[**Data sources**](http://docs.google.com/data_sources/index.html)

* [HTML (DOM) sourced data](http://docs.google.com/data_sources/dom.html)
* [Ajax sourced data](http://docs.google.com/data_sources/ajax.html)
* [Javascript sourced data](http://docs.google.com/data_sources/js_array.html)
* [Server-side processing](http://docs.google.com/data_sources/server_side.html)

[**API**](http://docs.google.com/api/index.html)

* [Add rows](http://docs.google.com/api/add_row.html)
* [Individual column searching (text inputs)](http://docs.google.com/api/multi_filter.html)
* [Individual column searching (select inputs)](http://docs.google.com/api/multi_filter_select.html)
* [Highlighting rows and columns](http://docs.google.com/api/highlight.html)
* [Child rows (show extra / detailed information)](http://docs.google.com/api/row_details.html)
* [Row selection (multiple rows)](http://docs.google.com/api/select_row.html)
* [Row selection and deletion (single row)](http://docs.google.com/api/select_single_row.html)
* [Form inputs](http://docs.google.com/api/form.html)
* [Index column](http://docs.google.com/api/counter_columns.html)
* [Show / hide columns dynamically](http://docs.google.com/api/show_hide.html)
* [Using API in callbacks](http://docs.google.com/api/api_in_init.html)
* [Scrolling and Bootstrap tabs](http://docs.google.com/api/tabs_and_scrolling.html)
* [Search API (regular expressions)](http://docs.google.com/api/regex.html)

[**Ajax**](http://docs.google.com/index.html)

* [Ajax data source (arrays)](http://docs.google.com/simple.html)
* [Ajax data source (objects)](http://docs.google.com/objects.html)
* [Nested object data (objects)](http://docs.google.com/deep.html)
* [Nested object data (arrays)](http://docs.google.com/objects_subarrays.html)
* [Orthogonal data](http://docs.google.com/orthogonal-data.html)
* [Generated content for a column](http://docs.google.com/null_data_source.html)
* [Custom data source property](http://docs.google.com/custom_data_property.html)
* [Flat array data source](http://docs.google.com/custom_data_flat.html)
* [Deferred rendering for speed](http://docs.google.com/defer_render.html)

[**Server-side**](http://docs.google.com/server_side/index.html)

* [Server-side processing](http://docs.google.com/server_side/simple.html)
* [Custom HTTP variables](http://docs.google.com/server_side/custom_vars.html)
* [POST data](http://docs.google.com/server_side/post.html)
* [Automatic addition of row ID attributes](http://docs.google.com/server_side/ids.html)
* [Object data source](http://docs.google.com/server_side/object_data.html)
* [Row details](http://docs.google.com/server_side/row_details.html)
* [Row selection](http://docs.google.com/server_side/select_rows.html)
* [JSONP data source for remote domains](http://docs.google.com/server_side/jsonp.html)
* [Deferred loading of data](http://docs.google.com/server_side/defer_loading.html)
* [Pipelining data to reduce Ajax calls for paging](http://docs.google.com/server_side/pipeline.html)

[**Plug-ins**](http://docs.google.com/plug-ins/index.html)

* [API plug-in methods](http://docs.google.com/plug-ins/api.html)
* [Ordering plug-ins (with type detection)](http://docs.google.com/plug-ins/sorting_auto.html)
* [Ordering plug-ins (no type detection)](http://docs.google.com/plug-ins/sorting_manual.html)
* [Custom filtering - range search](http://docs.google.com/plug-ins/range_filtering.html)
* [Live DOM ordering](http://docs.google.com/plug-ins/dom_sort.html)

Please refer to the [DataTables documentation](http://www.datatables.net) for full information about its API properties and methods.

Additionally, there are a wide range of [extensions](http://www.datatables.net/extensions) and [plug-ins](http://www.datatables.net/plug-ins) which extend the capabilities of DataTables.

DataTables designed and created by [SpryMedia Ltd](http://www.sprymedia.co.uk) © 2007-2016

DataTables is licensed under the [MIT license](http://www.datatables.net/mit).