# Mediatype constraints

## Summary

Mediatype constraints check the mediatype of the target resource. It is checked if the resource has one of the mediatypes XML, JSON or CSV. In case of CSV, optional additional constraints refer to the number of rows and the numbers of columns.

## Element

<mediatype>

## Examples

<mediatype eq="json"/>

<mediatype eq="xml json"/>

<mediatype eq="csv"/>

<mediatype eq="csv" csv.separator="semicolon" csv.withHeader="true"/>  
<mediatype eq="csv" csv.separator="semicolon" csv.header="yes"   
 csv.columnCount="71" csv.columnCountMsg="Not the expected number of columns"   
 csv.rowCount="249" csv.rowCountMsg="Not the expected number of rows"/>

## Example schemas

See:

$greenfox/example-schemas/constraint-components/Mediatype

## Constraint nodes

The constraints are specified by attributes on <mediatype>, summarized in the following table.

**Table 1**: Constraint nodes supported by Mediatype constraints. All nodes are attributes on the <mediatype> element.

|  |  |  |  |
| --- | --- | --- | --- |
| **Check node** | **Expectation** | **Example** | **Notes** |
| @eq | The resource has one of the mediatypes given by the attribute value | eq="xml"  eq="xml json" | Single mediatype or whitespace-separated list of mediatypes; the resource must have one of the mediatypes |
| @csv.columnCount | Number of csv columns | csv.columnCount="4" | All rows must have this number of columns |
| @csv.minColumnCount | Minimum number of csv columns | csv.minColumnCount="4" | Only when checking for mediatype csv: in all rows the number of columns must be >= the attribute value; the constraint is likely to detect an unexpected column separator |
| @csv.maxColumnCount | Maximum number of csv columns | csv.maxColumnCount="4" | Only when checking for mediatype csv: in all rows the number of columns must be <= the attribute value |
| @csv.rowCount | Number of csv rows | csv.rowCount="12" | Only when checking for mediatype csv: the number of rows must be equal to the attribute value; if @csv.withHeader is yes, the row count does not include the first row |
| @csv.minRowCount | Minimum number of csv rows | csv.minRowCount=”12” | Only when checking for mediatype csv: the number of rows must be >= the attribute value; if @csv.withHeader is yes, the row count does not include the first row |
| @csv.maxRowCount | Maximum number of csv rows | csv.maxRowCount="12" | Only when checking for mediatype csv: the number of rows must be <= the attribute value; if @csv.withHeader is yes, the row count does not include the first row |

## Option nodes

There are several option nodes, all of which control the parsing of a resource into a CSV representation. The options are represented in the following table.

Note: all options have the semantics defined by BaseX ([BaseX#csv-module](https://docs.basex.org/wiki/CSV_Module)).

**Table 2**: The option nodes supported by Mediatype constraints. The nodes are attributes on the <mediatype> element.

|  |  |  |  |
| --- | --- | --- | --- |
| **Check node** | **Semantics** | **Example** | **Notes** |
| @csv.separator | The character separating the column values of a row | csv.separator="comma"  csv.separator="semicolon"  csv.separator="tab"  csv.separator="colon"  csv.separator="space"  csv.separator="~"  csv.separator="#"  csv.separator="\*" | Default: comma; the value is either one of five tokens (see examples), or an arbitrary single character |
| @csv.header | Indicates if the first line of the parsed CSV contains column headers | csv.header="yes" | Default: no; has an impact on the number of rows checked by @csv.minRowCount or @csv.maxRowCount – the number of rows does not include the header row |
| @csv.format | Controls the node tree representation of the CSV data;  value direct:  field names are element names;  value attributes: field names are conveyed by @name attributes;  value xquery:  data are represented as an XQuery map | csv.format="attributes" | Default: direct |
| @csv.lax | Indicates if a lax approach is used to convert QNames to JSON names | csv.lax="yes" | Default: yes |
| @csv.quotes | Indicates if quotes at the start and end of a value will be treated as control characters | csv.quotes="no" | Default: yes |
| @csv.backslashes | If yes, \r, \n and \t are replaced by control characters; if no, two consecutive quotes are replaced by a single quote | csv.backslashes="yes" | Default: no |

## Message nodes

For each constraint node (see table 1), two message nodes are supported – an error message and a message commenting a pass:

@eqMsg @eqMsgOK

@csv.columnCountMsg @csv.columnCountMsgOK

@csv.minColumnCountMsg @csv.minColumnCountMsgOK

@csv.maxColumnCountMsg @csv.maxColumnCountMsgOK

@csv.rowCountMsg @csv.rowCountMsgOK

@csv.minRowCountMsg @csv.minRowCountMsgOK

@csv.maxRowCountMsg @csv.maxRowCountMsgOK