(/contact)   EN
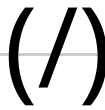
Docs (/guide)

(/)

Docs (/guide)

You are looking at documentation for an older release. Not what you want? See the current release documentation (../current/index.html).

Logstash Reference [5.3] (index.html) » Filter plugins (filter-plugins.html) » csv

CSV   edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

- Version: 3.0.2
- Released on: July 14, 2016
- Changelog (https://github.com/logstash-plugins/logstash-filter-csv/blob/master/CHANGELOG.md#302)

Getting Help   edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

For questions about the plugin, open a topic in the Discuss (http://discuss.elastic.co) forums. For bugs or feature requests, open an issue in Github (https://github.com/elastic/logstash). For the list of Elastic supported plugins, please consult the Elastic Support Matrix (https://www.elastic.co/support/matrix#show_logstash_plugins).

Description   edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

The CSV filter takes an event field containing CSV data, parses it, and stores it as individual fields (can optionally specify the names). This filter can also parse data with any separator, not just commas.

Synopsis   edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

This plugin supports the following configuration options:

Required configuration options:

```
csv {
}
```

Available configuration options:

Setting	Input type	Required
add_field (plugins-filters-csv.html#plugins-filters-csv-add_field)	hash (configuration-file-structure.html#hash)	No
add_tag (plugins-filters-csv.html#plugins-filters-csv-add_tag)	array (configuration-file-structure.html#array)	No
autogenerate_column_names (plugins-filters-csv.html#plugins-filters-csv-autogenerate_column_names)	boolean (configuration-file-structure.html#boolean)	No
columns (plugins-filters-csv.html#plugins-filters-csv-columns)	array (configuration-file-structure.html#array)	No
convert (plugins-filters-csv.html#plugins-filters-csv-convert)	hash (configuration-file-structure.html#hash)	No
enable_metric (plugins-filters-csv.html#plugins-filters-csv-enable_metric)	boolean (configuration-file-structure.html#boolean)	No
id (plugins-filters-csv.html#plugins-filters-csv-id)	string (configuration-file-structure.html#string)	No
periodic_flush (plugins-filters-csv.html#plugins-filters-csv-periodic_flush)	boolean (configuration-file-structure.html#boolean)	No
quote_char (plugins-filters-csv.html#plugins-filters-csv-quote_char)	string (configuration-file-structure.html#string)	No
remove_field (plugins-filters-csv.html#plugins-filters-csv-remove_field)	array (configuration-file-structure.html#array)	No
remove_tag (plugins-filters-csv.html#plugins-filters-csv-remove_tag)	array (configuration-file-structure.html#array)	No

Setting	Input type	Required
`separator` (plugins-filters-csv.html#plugins-filters-csv-separator)	string (configuration-file-structure.html#string)	No
`skip_empty_columns` (plugins-filters-csv.html#plugins-filters-csv-skip_empty_columns)	boolean (configuration-file-structure.html#boolean)	No
`source` (plugins-filters-csv.html#plugins-filters-csv-source)	string (configuration-file-structure.html#string)	No
`target` (plugins-filters-csv.html#plugins-filters-csv-target)	string (configuration-file-structure.html#string)	No

Details edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

add_field
edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

- Value type is hash (configuration-file-structure.html#hash)
- Default value is `{}`

If this filter is successful, add any arbitrary fields to this event. Field names can be dynamic and include parts of the event using the `%{field}`.

Example:

```
filter {
  csv {
    add_field => { "foo_%{somefield}" => "Hello world, from %{host}" }
  }
}
```

```
# You can also add multiple fields at once:
filter {
  csv {
    add_field => {
      "foo_%{somefield}" => "Hello world, from %{host}"
      "new_field" => "new_static_value"
    }
  }
}
```

If the event has field `"somefield" == "hello"` this filter, on success, would add field
`foo_hello` if it is present, with the value above and the `%{host}` piece replaced with that
value from the event. The second example would also add a hardcoded field.

add_tag
edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)
- Value type is array (configuration-file-structure.html#array)
- Default value is `[]`

If this filter is successful, add arbitrary tags to the event. Tags can be dynamic and include
parts of the event using the `%{field}` syntax.

Example:

```
filter {
  csv {
    add_tag => [ "foo_%{somefield}" ]
  }
}
```

```
# You can also add multiple tags at once:
filter {
  csv {
    add_tag => [ "foo_%{somefield}", "taggedy_tag"]
  }
}
```

If the event has field `"somefield" == "hello"` this filter, on success, would add a tag
`foo_hello` (and the second example would of course add a `taggedy_tag` tag).

autogenerate_column_names
edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)
- Value type is boolean (configuration-file-structure.html#boolean)
- Default value is `true`

Define whether column names should autogenerated or not. Defaults to true. If set to false,
columns not having a header specified will not be parsed.

columns
edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)
- Value type is array (configuration-file-structure.html#array)
- Default value is `[]`

Define a list of column names (in the order they appear in the CSV, as if it were a header
line). If `columns` is not configured, or there are not enough columns specified, the default
column names are "column1", "column2", etc. In the case that there are more columns in

the data than specified in this column list, extra columns will be auto-numbered: (e.g.
"user_defined_1", "user_defined_2", "column3", "column4", etc.)

convert

* Value type is hash (configuration-file-structure.html#hash)

* Default value is `{}`

Define a set of datatype conversions to be applied to columns. Possible conversions are
integer, float, date, date_time, boolean

Example:

```
filter {
  csv {
    convert => { "column1" => "integer", "column2" => "boolean" }
  }
}
```

enable_metric

* Value type is boolean (configuration-file-structure.html#boolean)

* Default value is `true`

Disable or enable metric logging for this specific plugin instance by default we record all the
metrics we can, but you can disable metrics collection for a specific plugin.

id

* Value type is string (configuration-file-structure.html#string)

* There is no default value for this setting.

Add a unique `ID` to the plugin configuration. If no ID is specified, Logstash will generate
one. It is strongly recommended to set this ID in your configuration. This is particularly
useful when you have two or more plugins of the same type, for example, if you have 2 grok
filters. Adding a named ID in this case will help in monitoring Logstash when using the
monitoring APIs.

```
output {
  stdout {
    id => "my_plugin_id"
  }
}
```

periodic_flush

- Value type is boolean (configuration-file-structure.html#boolean)

edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

- Default value is `false`

Call the filter flush method at regular interval. Optional.

quote_char

edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

- Value type is string (configuration-file-structure.html#string)

- Default value is `"\""`

Define the character used to quote CSV fields. If this is not specified the default is a double quote `"`. Optional.

remove_field

edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

- Value type is array (configuration-file-structure.html#array)

- Default value is `[]`

If this filter is successful, remove arbitrary fields from this event. Example:

```
filter {
  csv {
    remove_field => [ "foo_%{somefield}" ]
  }
}
```

```
# You can also remove multiple fields at once:
filter {
  csv {
    remove_field => [ "foo_%{somefield}", "my_extraneous_field" ]
  }
}
```

If the event has field `"somefield" == "hello"` this filter, on success, would remove the field with name `foo_hello` if it is present. The second example would remove an additional, non-dynamic field.

remove_tag

edit (https://github.com/logstash-plugins/logstash-filter-csv/edit/master/lib/logstash/filters/csv.rb)

- Value type is array (configuration-file-structure.html#array)

- Default value is `[]`

If this filter is successful, remove arbitrary tags from the event. Tags can be dynamic and include parts of the event using the `%{field}` syntax.

Example:

```
filter {
  csv {
    remove_tag => [ "foo_%{somefield}" ]
  }
}
```

```
# You can also remove multiple tags at once:
filter {
  csv {
    remove_tag => [ "foo_%{somefield}", "sad_unwanted_tag"]
  }
}
```

If the event has field `"somefield" == "hello"` this filter, on success, would remove the tag `foo_hello` if it is present. The second example would remove a sad, unwanted tag as well.

separator
- Value type is string (configuration-file-structure.html#string)
- Default value is `","`

Define the column separator value. If this is not specified, the default is a comma `,`. Optional.

skip_empty_columns
- Value type is boolean (configuration-file-structure.html#boolean)
- Default value is `false`

Define whether empty columns should be skipped. Defaults to false. If set to true, columns containing no value will not get set.

source
- Value type is string (configuration-file-structure.html#string)
- Default value is `"message"`

The CSV data in the value of the `source` field will be expanded into a data structure.

target
- Value type is string (configuration-file-structure.html#string)
- There is no default value for this setting.

Define target field for placing the data. Defaults to writing to the root of the event.

Top Videos

- Elasticsearch Demo (https://www.elastic.co/webinars/getting-started-elasticsearch?
  baymax=default&elektra=docs&storm=top-video)

- Kibana 101 (https://www.elastic.co/webinars/getting-started-kibana?
  baymax=default&elektra=docs&storm=top-video)

- Logstash Primer (https://www.elastic.co/webinars/getting-started-logstash?
  baymax=default&elektra=docs&storm=top-video)

On this page

Logstash Reference: 5.3   ▾
Logstash Introduction (introduction.html)
Getting Started with Logstash (getting-started-with-logstash.html)
How Logstash Works (pipeline.html)
Setting Up and Running Logstash (setup-logstash.html)
Breaking changes (breaking-changes.html)
Upgrading Logstash (upgrading-logstash.html)
Configuring Logstash (configuration.html)
Working with Filebeat Modules (filebeat-modules.html)
Deploying and Scaling Logstash (deploying-and-scaling.html)
Performance Tuning (performance-tuning.html)
Monitoring APIs (monitoring.html)
Working with plugins (working-with-plugins.html)
Input plugins (input-plugins.html)
Output plugins (output-plugins.html)