

Title: Store parsed logs in files Description: log transformation, log shipper, ETL, JSON, log management

Output Plugin: Files

Writes Logagent output into files. The file name could be dynamically generated using extracted field values in the template for the file name. Following file formats are supported:

- **ldjson** - Line delimited JSON
- **pretty** - Pretty JSON
- **yaml** - YAML format
- **template** - Any format using format templates in the **template** property

File rotation is supported via following properties:

- **size** - Maximum file size e.g. 10M (MegaBytes), 1G (GigaBytes), 300k (KiloBytes), 10000B (Bytes)
- **maxFiles** - Maximum number of files to keep on disk
- **interval** - Interval for file rotation e.g. 5s (seconds), 10m (minutes), '1d' (days), 1h (hours)
- **compress** - Boolean flag to enable compression of rotated files (true/false).

Please refer to rotating-file-stream for details on the file rotation options.

This plugin is particularly useful to export data from supported data sources like MySQL, Postgres, MS-SQL, Elasticsearch, Logfiles or Kafka topics to files.

Configuration

```
input:
  stdin: true
```

```
# example parser, extracting country name to route file output to files named {country}.txt
parser:
  patternFiles: []
  patterns:
    - sourceName: !!js/regexp /.*/
      match:
        - type: countryInfo
          regex: !!js/regexp /(.*?)\s(.*?)\s(.*?)\s/i
          fields:
            - country:string
            - anotherField:string
            - andAnotherField:string
```

```
output:
  files:
```

```
module: output-files
debug: false
fileName: '/tmp/{country}.txt'
# available format values: 'ldjson' (default), 'yaml', 'pretty' (pretty JSON), 'template'
format: template
# define a template for comma separated format
template: '{country},{anotherField},{andAnotherField}'
# matches context.sourceName == data.logSource
sourceName: .*
# matches data._type field
typeName: countryInfo
# properties for https://www.npmjs.com/search?q=rotating-file-stream
size: 10M
maxFiles: 2
interval: 1d
compress: true
```

Start logagent

```
logagent --config output-file.yaml
```