

# Package ‘sparkwarc’

January 11, 2022

**Type** Package

**Title** Load WARC Files into Apache Spark

**Version** 0.1.6

**Maintainer** Edgar Ruiz <edgar@rstudio.com>

**Description** Load WARC (Web ARChive) files into Apache Spark using 'sparklyr'. This allows to read files from the Common Crawl project <<http://commoncrawl.org/>>.

**License** Apache License 2.0

**BugReports** <https://github.com/r-spark/sparkwarc>

**Encoding** UTF-8

**Imports** DBI, sparklyr, Rcpp

**RoxygenNote** 7.1.1

**LinkingTo** Rcpp,

**SystemRequirements** C++11

**NeedsCompilation** yes

**Author** Javier Luraschi [aut],  
Yitao Li [aut] (<<https://orcid.org/0000-0002-1261-905X>>),  
Edgar Ruiz [aut, cre]

**Repository** CRAN

**Date/Publication** 2022-01-11 08:50:02 UTC

## R topics documented:

cc_warc . . . . .	2
rcpp_read_warc_sample . . . . .	2
sparkwarc . . . . .	3
spark_rcpp_read_warc . . . . .	3
spark_read_warc . . . . .	3
spark_read_warc_sample . . . . .	4
spark_warc_sample_path . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

`cc_warc`*Provides WARC paths for commoncrawl.org*

---

**Description**

Provides WARC paths for commoncrawl.org. To be used with `spark_read_warc`.

**Usage**

```
cc_warc(start, end = start)
```

**Arguments**

<code>start</code>	The first path to retrieve.
<code>end</code>	The last path to retrieve.

**Examples**

```
cc_warc(1)
cc_warc(2, 3)
```

---

`rcpp_read_warc_sample` *Loads the sample warc file in Rcpp*

---

**Description**

Loads the sample warc file in Rcpp

**Usage**

```
rcpp_read_warc_sample(filter = "", include = "")
```

**Arguments**

<code>filter</code>	A regular expression used to filter to each warc entry efficiently by running native code using Rcpp.
<code>include</code>	A regular expression used to keep only matching lines efficiently by running native code using Rcpp.

---

sparkwarc	<i>sparkwarc</i>
-----------	------------------

---

**Description**

Sparklyr extension for loading WARC Files into Apache Spark

---

spark_rcpp_read_warc	<i>Reads a WARC File into using Rcpp</i>
----------------------	--

---

**Description**

Reads a WARC (Web ARChive) file using Rcpp.

**Usage**

```
spark_rcpp_read_warc(path, match_warc, match_line)
```

**Arguments**

path	The path to the file. Needs to be accessible from the cluster. Supports the "hdfs://", "s3n://" and "file://" protocols.
match_warc	include only warc files mathcing this character string.
match_line	include only lines mathcing this character string.

---

spark_read_warc	<i>Reads a WARC File into Apache Spark</i>
-----------------	--

---

**Description**

Reads a WARC (Web ARChive) file into Apache Spark using sparklyr.

**Usage**

```
spark_read_warc(
  sc,
  name,
  path,
  repartition = 0L,
  memory = TRUE,
  overwrite = TRUE,
  match_warc = "",
  match_line = "",
  parser = c("r", "scala"),
  ...
)
```

**Arguments**

sc	An active spark_connection.
name	The name to assign to the newly generated table.
path	The path to the file. Needs to be accessible from the cluster. Supports the "hdfs://", "s3n://" and "file://" protocols.
repartition	The number of partitions used to distribute the generated table. Use 0 (the default) to avoid partitioning.
memory	Boolean; should the data be loaded eagerly into memory? (That is, should the table be cached?)
overwrite	Boolean; overwrite the table with the given name if it already exists?
match_warc	include only warc files matching this character string.
match_line	include only lines matching this character string.
parser	which parser implementation to use? Options are "scala" or "r" (default).
...	Additional arguments reserved for future use.

**Examples**

```
## Not run:
library(sparklyr)
library(sparkwarc)
sc <- spark_connect(master = "local")
sdf <- spark_read_warc(
  sc,
  name = "sample_warc",
  path = system.file(file.path("samples", "sample.warc"), package = "sparkwarc"),
  memory = FALSE,
  overwrite = FALSE
)

spark_disconnect(sc)

## End(Not run)
```

---

```
spark_read_warc_sample
```

*Loads the sample warc file in Spark*

---

**Description**

Loads the sample warc file in Spark

**Usage**

```
spark_read_warc_sample(sc, filter = "", include = "")
```

**Arguments**

sc	An active spark_connection.
filter	A regular expression used to filter to each warc entry efficiently by running native code using Rcpp.
include	A regular expression used to keep only matching lines efficiently by running native code using Rcpp.

---

spark\_warc\_sample\_path

*Retrieves sample warc path*

---

**Description**

Retrieves sample warc path

**Usage**

spark\_warc\_sample\_path()

# Index

`cc_warc`, [2](#)

`rcpp_read_warc_sample`, [2](#)

`spark_rcpp_read_warc`, [3](#)

`spark_read_warc`, [3](#)

`spark_read_warc_sample`, [4](#)

`spark_warc_sample_path`, [5](#)

`sparkwarc`, [3](#)