Package: rtika (via r-universe)

October 29, 2026

Type Package

Title R Interface to 'Apache Tika'

Version 3.2.3

Maintainer Sasha Goodman <goodmansasha@gmail.com>

Imports curl, sys (>= 2.1), stats, utils, digest, backports

Suggests jsonlite, xml2, testthat, knitr, rmarkdown, covr, magrittr

License Apache License 2.0 | file LICENSE

SystemRequirements Java (>=11)

Description Extract text or metadata from over a thousand file types, using Apache Tika <https://tika.apache.org/>. Get either plain text or structured XHTML content.

Depends R (>= 3.5.0)

Encoding UTF-8

RoxygenNote 7.3.3

URL https://docs.ropensci.org/rtika/, https://github.com/ropensci/rtika/

BugReports https://github.com/ropensci/rtika/issues/

VignetteBuilder knitr

Config/pak/sysreqs default-jdk libssl-dev

Repository https://ropensci.r-universe.dev

Date/Publication 2025-10-12 13:59:05 UTC

RemoteUrl https://github.com/ropensci/rtika

RemoteRef HEAD

RemoteSha 350fcc3e90ab83dad8770021d6258147cf0c2ff0

Contents

install_tika 2
java 3
tika 4
tika_check 7
tika_fetch 8
tika_html 9
tika_jar 10
tika_json 10
tika_json_text 11
tika_text 12
tika_xml 13

Index **14**

`install_tika` *Install or Update the Apache Tika* `jar`

Description

This downloads and installs the Tika App `jar` (~55 MB) into a user directory, and verifies the integrity of the file using a checksum. The default settings should work fine.

Usage

```
install_tika(
  version = "3.2.3",
 digest = paste0("5908760c8506d6d1f2c37b59b21b513c2451960b471dde0b353d0ac825a3ea6a",
    "809714e0651de9883f8257444b2a8c56c6982a55fed74b5e5840261aa893b650"),
  mirrors = c("https://ftp.wayne.edu/apache/tika/",
   "http://mirrors.ocf.berkeley.edu/apache/tika/", "http://apache.cs.utah.edu/tika/",
    "http://mirror.cc.columbia.edu/pub/software/apache/tika/"),
  retries = 2,
  url = character()
)
```

Arguments

`version`	The declared Tika version
`digest`	The sha512 checksum. Set to an empty string `""` to skip the check.
`mirrors`	A vector of Apache mirror sites. One is picked randomly.
`retries`	The number of times to try the download.
`url`	Optional url to a particular location of the tika app. Setting this to any character string overrides downloading from random mirrors.

Value

Logical if the installation was successful.

Details

The default settings of `install_tika()` should typically be left as they are.

This function will download the version of the Tika `jar` tested to work with this package, and can verify file integrity using a checksum.

It will normally download from a random Apache mirror. If the mirror fails, it tries the archive at `http://archive.apache.org/dist/tika/`. You can also enter a value for `url` directly to override this.

It will download into a directory determined by `tools::R_user_dir("rtika", which = "data")`, specific to the operating system.

If `tika()` is stopping with an error compalining about the `jar`, try running `install_tika()` again.

Uninstalling

If you are uninstalling the entire `rtika` package and want to remove the Tika App `jar` also, run:

`unlink(tools::R_user_dir("rtika", which = "data"), recursive = TRUE)`

Alternately, navigate to the install folder and delete it manually. It is the file path returned by `tools::R_user_dir("rtika", which = "data")`. The path is OS specific.

Distribution

Tika is distributed under the Apache License Version 2.0, which generally permits distribution of the code "Object" without the "Source". The master copy of the Apache Tika source code is held in GIT. You can fetch (clone) the large source from GitHub (https://github.com/apache/tika).

Examples

```
if (interactive()) {
  install_tika()
}
```

---

java *System Command to Run Java*

Description

Gets the system command needed to run Java from the command line, as a string. Typically, this is the string: 'java'. However, if the R session has the `JAVA_HOME` environmental variable set, it will use that to locate java instead. This can be persisted over sessions (see the Details below).

Usage

```
java()
```

Value

The system command needed to invoke Java, as a string.

Details

This function is used by all of the `tika()` functions internally as the default value to its `java` parameter.

This function tries to find an environmental variable using `Sys.getenv("JAVA_HOME")`. It looks for the `java` executable inside the `bin` directory in the `JAVA_HOME` directory.

If you want to use a specific version of Java, set the `JAVA_HOME` variable using `Sys.setenv(JAVA_HOME = 'my path')`, where 'my path' is the path to a folder that has a `bin` directory with a `java` executable.

For example, on Windows 11 `JAVA_HOME` might be `C:/Program Files/Java/jdk-21`. On Ubuntu and macOS, it might be something like `/usr/lib/jvm/temurin-21-jdk`.

The `JAVA_HOME` variable can also be set to persist over sessions. Add the path to the `.Rprofile` by adding `Sys.setenv(JAVA_HOME = 'my path')`, and it will use that every time R is started.

Examples

```
# Typically, this function returns the string 'java'.
# If JAVA_HOME is set, it's a path to java in a 'bin' folder.
java()
```

`tika` *Main R Interface to 'Apache Tika'*

Description

Extract text or metadata from over a thousand file types. Get either plain text or structured `XHTML`. Metadata includes `Content-Type`, character encoding, and Exif data from jpeg or tiff images. See the long list of supported file types, click the "Supported Formats" link on this page : https://tika.apache.org/.

Usage

```
tika(
  input,
  output = c("text", "jsonRecursive", "xml", "html")[1],
  output_dir = "",
  return = TRUE,
  java = rtika::java(),
```

```
  jar = rtika::tika_jar(),
  threads = 2,
  max_restarts = integer(),
  timeout = 3e+05,
  max_file_size = integer(),
  config = system.file("extdata", "ocr.xml", package = "rtika"),
  args = character(),
  quiet = TRUE,
  cleanup = TRUE,
  lib.loc = .libPaths()
)
```

Arguments

input	Character vector describing the paths to the input documents. Strings starting with 'http://','https://', or 'ftp://' are downloaded to a temporary directory. On Windows, the local paths cannot span drives because of a Windows convention.
output	Optional character vector of the output format. The default, "text", gets plain text without metadata. "xml" and "html" get XHTML text with metadata. "jsonRecursive" gets XHTML text and json metadata. c("jsonRecursive","text") or c("J","t") get plain text and json metadata. See the 'Output Details' section.
output_dir	Optional directory path to save the converted files in. Tika may overwrite files so an empty directory is best. See the 'Output Details' section before using.
return	Logical if an R object should be returned. Defaults to TRUE. If set to FALSE, and output_dir (above) must be specified.
java	Optional command to invoke Java. For example, it can be the full path to a particular Java version. See the Configuration section below.
jar	Optional alternative path to a tika-app-X.XX.jar. Useful if this package becomes out of date.
threads	Integer of the number of file consumer threads Tika uses. Defaults to 2.
max_restarts	Integer of the maximum number of times the watchdog process will restart the child process. The default is no limit.
timeout	Integer of the number of milliseconds allowed to a parse before the process is killed and restarted. Defaults to 300000.
max_file_size	Integer of the maximum bytes allowed. Do not process files larger than this. The default is unlimited.
config	Path to the XML config file. Defaults to system.file("extdata", "ocr.xml", package = "rtika")'. There is also a no-ocr.xml file available.
args	Optional character vector of additional arguments passed to Tika, that may not yet be implemented in this R interface, in the pattern of c('-arg1','setting1','-arg2','setting2').
quiet	Logical if Tika command line messages and errors are to be suppressed. Defaults to TRUE.
cleanup	Logical to clean up temporary files after running the command, which can accumulate. Defaults to TRUE. They are in tempdir(). These files are automatically removed at the end of the R session even if set to FALSE.

lib.loc — Optional character vector describing the library paths. Normally, it's best to leave this parameter alone. The parameter is included mainly for package testing.

Value

A character vector in the same order and with the same length as input. Unprocessed files are as.character(NA). If return = FALSE, then a NULL value is invisibly returned. See the Output Details section below.

Output Details

If an input file did not exist, could not be downloaded, was a directory, or Tika could not process it, the result will be as.character(NA) for that file.

By default, output = "text" and this produces plain text with no metadata. Some formatting is preserved in this case using tabs, newlines and spaces.

Setting output to either "xml" or the shortcut "x" will produce a strict form of HTML known as XHTML, with metadata in the head node and formatted text in the body. Content retains more formatting with "xml". For example, a Word or Excel table will become a HTML table, with table data as text in td elements. The "html" option and its shortcut "h" seem to produce the same result as "xml". Parse XHTML output with xml2::read_html.

Setting output to "jsonRecursive" or its shortcut "J" produces a tree structure in 'json'. Metadata fields are at the top level. The XHTML or plain text will be found in the X-TIKA:content field. By default the text is XHTML. This can be changed to plain text like this: output=c("jsonRecursive","text") or output=c("J","t"). This syntax is meant to mirror Tika's. Parse json with jsonlite::fromJSON.

If output_dir is specified, then the converted files will also be saved to this directory. It's best to use an empty directory because Tika may overwrite existing files. Tika seems to add an extra file extension to each file to reduce the chance, but it's still best to use an empty directory. The file locations within the output_dir maintain the same general path structure as the input files. Downloaded files have a path similar to the 'tempdir()' that R uses. The original paths are now relative to output_dir. Files are appended with .txt for the default plain text, but can be .json, .xml, or .html depending on the output setting. One way to get a list of the processed files is to use list.files with recursive=TRUE. If output_dir is not specified, files are saved to a volatile temp directory named by tempdir() and will be deleted when R shuts down. If this function will be run on very large batches repeatedly, these temporary files can be cleaned up every time by adding cleanup=TRUE.

Background

Tika is a foundational library for several Apache projects such as the Apache Solr search engine. It has been in development since at least 2007. The most efficient way I've found to process many thousands of documents is Tika's 'batch' mode, which is the only mode used in 'rtika'. There are potentially more things that can be done, given enough time and attention, because Apache Tika includes many libraries and methods in its .jar file. The source is available at: https://tika.apache.org/.

Installation

Tika requires Java 11 or later.

Java installation instructions are at https://openjdk.org/install/ or https://www.java.com/en/download/help/download_options.

By default, this R package internally invokes Java by calling the `java` command from the command line. To specify the path to a particular Java version, set the path in the `java` attribute of the `tika` function.

Examples

```
#extract text
batch <- c(
  system.file("extdata", "jsonlite.pdf", package = "rtika"),
  system.file("extdata", "curl.pdf", package = "rtika"),
  system.file("extdata", "table.docx", package = "rtika"),
  system.file("extdata", "xml2.pdf", package = "rtika"),
  system.file("extdata", "R-FAQ.html", package = "rtika"),
  system.file("extdata", "calculator.jpg", package = "rtika"),
  system.file("extdata", "tika.apache.org.zip", package = "rtika")
)
text = tika(batch)
cat(substr(text[1],45,450))

#more complex metadata
if(requireNamespace('jsonlite')){

  json = tika(batch,c('J','t'))
  # 'J' is shortcut for jsonRecursive
  # 't' for text
  metadata = lapply(json, jsonlite::fromJSON )

  #embedded resources
  lapply(metadata, function(x){ as.character(x$'Content-Type') })

  lapply(metadata, function(x){ as.character(x$'Creation-Date') })

  lapply(metadata, function(x){  as.character(x$'X-TIKA:embedded_resource_path') })
}
```

`tika_check` *Check Tika against a checksum*

Description

This is used by `install_tika()` internally, or can be called directly on a `jar` file. The latest `jar` files and checksums are at https://tika.apache.org/download.html.

Usage

```
tika_check(digest, jar = tika_jar(), algo = "sha512")
```

Arguments

digest	Character vector of length one with the target checksum.
jar	Optional alternative path to a Tika jar file.
algo	Optional algorithm used to create checksum. Defaults to SHA512.

Value

logical if the jar checksum matches digest.

---

tika_fetch *Fetch Files with the Content-Type Preserved in the File Extension*

Description

On the Internet, Content-Type information is mainly communicated via the server's headers. This is an issue if a file is saved to disk without examining the headers. The file can have a missing or incorrect file extension. For example, a URL ending in a slash (/) can produce file with the Content-Type of text/html. The same URL might also produce a image/jpeg or application/pdf file. URLs ending in .php, .cfm can produce any Content-Type. The downloaded file will lose the server's declared Content-Type unless its appended as a file extension. tika_fetch() gets a file from the URL, examines the server headers, and appends the matching file extension from Tika's database.

Usage

```
tika_fetch(
  urls,
  download_dir = tempdir(),
  ssl_verifypeer = TRUE,
  retries = 1,
  quiet = TRUE
)
```

Arguments

urls	Character vector of one or more URLs to be downloaded.
download_dir	Character vector of length one describing the path to the directory to save the results.
ssl_verifypeer	Logical, with a default of TRUE. Some server SSL certificates might not be recognized by the host system, and in these rare cases the user can ignore that if they know why.
retries	Integer of the number of times to retry each url after a failure to download.
quiet	Logical if download warnings should be printed. Defaults to FALSE.

Value

Character vector of the same length and order as input with the paths describing the locations of the downloaded files. Errors are returned as NA.

Examples

```
tika_fetch('https://tika.apache.org/')
# a unique file name with .html appended to it
```

`tika_html` *Get Structured XHTML*

Description

If `output_dir` is specified, files will have the `.html` file extension.

Usage

```
tika_html(input, ...)
```

Arguments

`input`	Character vector describing the paths and/or urls to the input documents.
`...`	Other parameters to be sent to `tika()`.

Value

A character vector in the same order and with the same length as `input`, of unparsed `XHTML`. Unprocessed files are `as.character(NA)`.

Examples

```
batch <- c(
 system.file("extdata", "jsonlite.pdf", package = "rtika"),
 system.file("extdata", "curl.pdf", package = "rtika"),
 system.file("extdata", "table.docx", package = "rtika"),
 system.file("extdata", "xml2.pdf", package = "rtika"),
 system.file("extdata", "R-FAQ.html", package = "rtika"),
 system.file("extdata", "calculator.jpg", package = "rtika"),
 system.file("extdata", "tika.apache.org.zip", package = "rtika")
)
html <- tika_html(batch)
```

tika_jar *Path to Apache Tika*

Description

Gets the path to the Tika App `.jar` installed by `tika_install()`.

Usage

```
tika_jar()
```

Value

A string describing the file path to the Tika App `.jar` file. If not found, `NA`.

Details

The `tika_jar()` function also checks if the `.jar` is actually on the file system.

The file path is used by all of the `tika()` functions by default.

Alternative Uses

You can call Apache Tika directly, as shown in the examples here.

It is better to use the `sys` package and avoid `system2()`, which has caused erratic, intermittent errors with Tika.

Examples

```
jar <- tika_jar()
# see help
sys::exec_wait('java',c('-jar',jar, '--help'))
# detect language of web page
sys::exec_wait('java',c('-jar',jar, '--language','https://tika.apache.org/'))
```

tika_json *Get json Metadata and XHTML Content*

Description

Tika can parse and extract text from almost anything, including zip, tar, tar.bz2, and other archives that contain documents. If you have a zip file with 100 text files in it, you can get the text and metadata for each file nested inside of the zip file. This recursive output is currently used for the jsonified mode. See: https://wiki.apache.org/tika/RecursiveMetadata

The document content is XHTML in the "X-TIKA:content" field.

If `output_dir` is specified, files will have the `.json` file extension.

Usage

```
tika_json(input, ...)
```

Arguments

input	Character vector describing the paths and/or urls to the input documents.
...	Other parameters to be sent to tika().

Value

A character vector in the same order and with the same length as input, of unparsed json. Unprocessed files are as.character(NA).

Examples

```
batch <- c(
 system.file("extdata", "jsonlite.pdf", package = "rtika"),
 system.file("extdata", "curl.pdf", package = "rtika"),
 system.file("extdata", "table.docx", package = "rtika"),
 system.file("extdata", "xml2.pdf", package = "rtika"),
 system.file("extdata", "R-FAQ.html", package = "rtika"),
 system.file("extdata", "calculator.jpg", package = "rtika"),
 system.file("extdata", "tika.apache.org.zip", package = "rtika")
)
json <- tika_json(batch)
```

---

tika_json_text *Get json Metadata and Plain Text Content*

Description

Tika can parse and extract text from almost anything, including zip, tar, tar.bz2, and other archives that contain documents. If you have a zip file with 100 text files in it, you can get the text and metadata for each file nested inside of the zip file. This recursive output is currently used for the jsonified mode. See: https://wiki.apache.org/tika/RecursiveMetadata

The document contents are plain text in the "X-TIKA:content" field.

If output_dir is specified, files will have the .json file extension.

Usage

```
tika_json_text(input, ...)
```

Arguments

input	Character vector describing the paths and/or urls to the input documents.
...	Other parameters to be sent to tika().

Value

A character vector in the same order and with the same length as `input`, of unparsed `json`. Unprocessed files are `as.character(NA)`.

Examples

```
batch <- c(
 system.file("extdata", "jsonlite.pdf", package = "rtika"),
 system.file("extdata", "curl.pdf", package = "rtika"),
 system.file("extdata", "table.docx", package = "rtika"),
 system.file("extdata", "xml2.pdf", package = "rtika"),
 system.file("extdata", "R-FAQ.html", package = "rtika"),
 system.file("extdata", "calculator.jpg", package = "rtika"),
 system.file("extdata", "tika.apache.org.zip", package = "rtika")
)
json <- tika_json_text(batch)
```

---

tika_text *Get Plain Text*

Description

If `output_dir` is specified, files will have the `.txt` file extension.

Usage

```
tika_text(input, ...)
```

Arguments

`input`	Character vector describing the paths and/or urls to the input documents.
`...`	Other parameters to be sent to `tika()`.

Value

A character vector in the same order and with the same length as `input`, of plain text. Unprocessed files are `as.character(NA)`.

Examples

```
batch <- c(
 system.file("extdata", "jsonlite.pdf", package = "rtika"),
 system.file("extdata", "curl.pdf", package = "rtika"),
 system.file("extdata", "table.docx", package = "rtika"),
 system.file("extdata", "xml2.pdf", package = "rtika"),
 system.file("extdata", "R-FAQ.html", package = "rtika"),
 system.file("extdata", "calculator.jpg", package = "rtika"),
 system.file("extdata", "tika.apache.org.zip", package = "rtika")
```

```
)
text <- tika_text(batch)
```

---

tika_xml *Get a Structured XHTML Rendition*

Description

If output_dir is specified, files will have the .xml file extension.

Usage

```
tika_xml(input, ...)
```

Arguments

input	Character vector describing the paths and/or urls to the input documents.
...	Other parameters to be sent to tika().

Value

A character vector in the same order and with the same length as input, of unparsed XHTML. Unprocessed files are as.character(NA).

Examples

```
batch <- c(
 system.file("extdata", "jsonlite.pdf", package = "rtika"),
 system.file("extdata", "curl.pdf", package = "rtika"),
 system.file("extdata", "table.docx", package = "rtika"),
 system.file("extdata", "xml2.pdf", package = "rtika"),
 system.file("extdata", "R-FAQ.html", package = "rtika"),
 system.file("extdata", "calculator.jpg", package = "rtika"),
 system.file("extdata", "tika.apache.org.zip", package = "rtika")
)
xml <- tika_xml(batch)
```

Index

install_tika, 2

java, 3

tika, 4
tika_check, 7
tika_fetch, 8
tika_html, 9
tika_jar, 10
tika_json, 10
tika_json_text, 11
tika_text, 12
tika_xml, 13