Formularbeginn

Formularende

**Introduction**

This page is documentation on tika's JSR 311 network server, tika-server. The server package uses the [Apache CXF](http://cxf.apache.org/) framework that provides an implementation of JAX-RS for Java. The Tika server component builds to a standalone package in Tika, tika-server.

**Installation**

To install:

1. Download the latest stable source from the [Apache Tika download page](http://tika.apache.org/download.html) or retrieve the latest code from [Github](https://github.com/apache/tika),
2. Build source using Maven,
3. Run the Apache Tika JAXRS server.

wget http://mirror.vorboss.net/apache/tika/tika-x.x-src.zip

unzip tika-x.x-src

cd ./tika-x.x/

mvn install

cd ./tika-server/target/

java -jar tika-server-x.x.jar

Remember to replace x.x with the version you have downloaded.

You will then see a message such as the following:

$ java -jar target/tika-server-1.2-SNAPSHOT.jar

Apr 4, 2012 7:48:49 AM org.apache.tika.server.TikaServerCli main

INFO: Starting Tikaserver ${project.version}

Apr 4, 2012 7:48:50 AM org.apache.cxf.endpoint.ServerImpl initDestination

INFO: Setting the server's publish address to be http://localhost:9998/

2012-04-04 07:48:50.316:INFO:oejs.Server:jetty-7.x.y-SNAPSHOT

2012-04-04 07:48:50.375:INFO:oejs.AbstractConnector:Started SelectChannelConnector@localhost:9998 STARTING

2012-04-04 07:48:50.399:INFO:oejsh.ContextHandler:started o.e.j.s.h.ContextHandler{,null}

Apr 4, 2012 7:48:50 AM org.apache.tika.server.TikaServerCli main

INFO: Started

Which lets you know that it started correctly.

You can specify additional information to change the host name and port number:

java -jar tika-server-x.x.jar --host=intranet.local --port=12345

Below is some basic documentation on how to interact with the services using cURL and HTTP.

**Services**

All services that take files use HTTP "PUT" requests. Original file must be sent in request body without any additional encoding (do not use multipart/form-data or other containers).

Information services (eg defined mimetypes, defined parsers etc) work with HTML "GET" requests.

You may optionally specify content type in "Content-Type" header. If you do not specify mime type, Tika will use its detectors to guess it.

You may specify additional identifier in URL after resource name, like "/tika/my-file-i-sent-to-tika-resource" for "/tika" resource. Tikaserver uses this name only for logging, so you may put there file name, UUID or any other identifier (do not forget to url-encode any special characters).

Resources may return following HTTP codes:

* 200 Ok - request completed sucessfully
* 204 No content - request completed sucessfully, result is empty
* 422 Unprocessable Entity - Unsupported mime-type, encrypted document & etc
* 500 Error - Error while processing document

**Metadata Resource**

/meta

HTTP PUTs a document to the /meta service and you get back "text/csv" of the metadata.

Some Example calls with cURL:

$ curl -X PUT --data-ascii @zipcode.csv http://localhost:9998/meta --header "Content-Type: text/csv"

$ curl -T price.xls http://localhost:9998/meta

Returns:

"Content-Encoding","ISO-8859-2"

"Content-Type","text/plain"

Get metadata as JSON:

$ curl -T test\_recursive\_embedded.docx http://localhost:9998/meta --header "Accept: application/json"

Or XMP:

$ curl -T test\_recursive\_embedded.docx http://localhost:9998/meta --header "Accept: application/rdf+xml"

Get specific metadata key's value as simple text string:

$ curl -T test\_recursive\_embedded.docx http://localhost:9998/meta/Content-Type --header "Accept: text/plain"

Returns:

application/vnd.openxmlformats-officedocument.wordprocessingml.document

Get specific metadata key's value(s) as CSV:

$ curl -T test\_recursive\_embedded.docx http://localhost:9998/meta/Content-Type --header "Accept: text/csv"

Or JSON:

$ curl -T test\_recursive\_embedded.docx http://localhost:9998/meta/Content-Type --header "Accept: application/json"

Or XMP:

$ curl -T test\_recursive\_embedded.docx http://localhost:9998/meta/Content-Type --header "Accept: application/rdf+xml"

**Note: when requesting specific metadata keys value(s) in XMP, make sure to request the XMP name, e.g. "dc:creator" vs. "Author"**

**Tika Resource**

/tika

HTTP PUTs a document to the /tika service and you get back the extracted text. HTTP GET prints a greeting stating the server is up.

Some Example calls with cURL:

**Get HELLO message back**

$ curl -X GET http://localhost:9998/tika

This is Tika Server. Please PUT

**Get the Text of a Document**

$ curl -X PUT --data-binary @GeoSPARQL.pdf http://localhost:9998/tika --header "Content-type: application/pdf"

$ curl -T price.xls http://localhost:9998/tika --header "Accept: text/html"

$ curl -T price.xls http://localhost:9998/tika --header "Accept: text/plain"

**Detector Resource**

/detect/stream

HTTP PUTs a document and uses the Default Detector from Tika to identify its MIME/media type. The caveat here is that providing a hint for the filename can increase the quality of detection.

Default return is a string of the Media type name.

Some Example calls with cURL:

**PUT an RTF file and get back RTF**

$ curl -X PUT --data-binary @TODO.rtf http://localhost:9998/detect/stream

**PUT a CSV file without filename hint and get back text/plain**

$ curl -X PUT --upload-file foo.csv http://localhost:9998/detect/stream

**PUT a CSV file with filename hint and get back text/csv**

$ curl -X PUT -H "Content-Disposition: attachment; filename=foo.csv" --upload-file foo.csv http://localhost:9998/detect/stream

**Recursive Metadata and Content**

/rmeta

Returns a JSONified list of Metadata objects for the container document and all embedded documents. The text that is extracted from each document is stored in the metadata object under "X-TIKA:content".

$ curl -T test\_recursive\_embedded.docx http://localhost:9998/rmeta

Returns:

[

{"Application-Name":"Microsoft Office Word",

"Application-Version":"15.0000",

"X-Parsed-By":["org.apache.tika.parser.DefaultParser","org.apache.tika.parser.microsoft.ooxml.OOXMLParser"],

"X-TIKA:content":"embed\_0 "

...

},

{"Content-Encoding":"ISO-8859-1",

"Content-Length":"8",

"Content-Type":"text/plain; charset=ISO-8859-1"

"X-TIKA:content":"embed\_1b",

...

}

...

]

**Unpack Resource**

/unpack

HTTP PUTs an embedded document type to the /unpack service and you get back a zip or tar of the extracted text for each resource filename in the original PUT embedded document type. You can also use /unpack/all to get back both the text and metadata.

Default return type is ZIP (without internal compression). Use "Accept" header for TAR return type.

Please note the mapping of this resource was changed in Apache Tika 1.6 from /unpacker/id to /unpack/id /all/id & /unpack/all/id ([TIKA-1324](https://issues.apache.org/jira/browse/TIKA-1324)).

Some Example calls with cURL:

**PUT zip file and get back met file zip**

$ curl -X PUT --data-binary @foo.zip http://localhost:9998/unpack --header "Content-type: application/zip"

**PUT doc file and get back met file tar**

$ curl -T Doc1\_ole.doc -H "Accept: application/x-tar" http://localhost:9998/unpack > /var/tmp/x.tar

**PUT doc file and get back the content and metadata**

$ curl -T Doc1\_ole.doc http://localhost:9998/unpack/all > /var/tmp/x.zip

Text is stored in \_\_TEXT\_\_ file, metadata cvs in \_\_METADATA\_\_. Use "accept" header if you want TAR output.

**Information Services**

**Defined Mime Types**

/mime-types

Mime types, their aliases, their supertype, and the parser. Available as plain text, json or human readable HTML

**Available Detectors**

/detectors

The top level Detector to be used, and any child detectors within it. Available as plain text, json or human readable HTML

**Available Parsers**

*TODO*

**Extracting A Document From A URL**

It is possible to use a remote file with TikaJAXRS by downloading it via its URL first then piping it to the appropriate service:

$ curl -s "http://url/to/my.file" | curl -X PUT -T - http://localhost:9998/meta

$ curl -s "http://url/to/my.file" | curl -X PUT -T - http://localhost:9998/tika

The caveat with above is that it fetches the entire file, so large files such as video can take some time to download. Therefore, you may wish to use curl to get preliminary information (content type, name and size) about the file before you proceed:

$ curl -I http://url/to/my.file

If the file should be parsed (E.g. you only want to get information about mp3s, mp4s and PDFs), send it on to TikaJAXRS.

TikaJAXRS (last edited 2014-12-29 22:32:54 by [DaveMeikle](http://wiki.apache.org/tika/DaveMeikle))

* Immutable Page
* [Info](http://wiki.apache.org/tika/TikaJAXRS?action=info)
* [Attachments](http://wiki.apache.org/tika/TikaJAXRS?action=AttachFile)

Formularbeginn

Formularende

* [MoinMoin Powered](http://moinmo.in/)
* [Python Powered](http://moinmo.in/Python)
* [GPL licensed](http://moinmo.in/GPL)
* [Valid HTML 4.01](http://validator.w3.org/check?uri=referer)