Okapi

Category	Import Converter
Purpose	Enables Translate5 to import other file formats as just bilingual files.
Bootstrap Class	editor_Plugins_Okapi_Init
Type	Core plug-in (delivered with translate5 core)

Description

The Okapi plugin enables you to use Okapi Longhorn REST server for converting native source file formats like MS Office 2007, Indesign, HTML, XML, etc. to Xliff and thus directly import these native file formats in translate5.

The end user of translate5 has no contact with Okapi - everything is handled by translate5 on import and re-export after translation.

Here you find the list of all file formats supported by Okapi and thus supported by translate5 through the translate5 Okapi plugin.

Additional installation instructions

Download and install Okapi Longhorn as shown here: http://okapiframework.org/wiki/index.php?title=Longhorn#Download_and_Installation.

If you have a working and running Tomcat that has autoDeploy set to "true" in the /etc/tomcat9/server.xml file, than to deploy your okapi-longhorn.war file you can simply copy it to the webapps-folder of your Tomcat installation (on Ubuntu this is located under /var/lib/tomcat9/webapps). You may have to set the file access rights of the war file, so that Tomcat can handle the file and you may need to restart Tomcat.

Important hints for OpenJDK

OpenJDK is the Java runtime usually used on Linux for Tomcat. Since May 2022 there is an OpoenJDK update for most java versions, where new limits for certain xpath recursions are introduced.

This leads to problems, when using Okapi with the XML ITS filter.

Therefore you should add the following increased limit to the startup java settings of your tomcat:

-Djdk.xml.xpathExprOpLimit=200

For doing this, open the setenv.sh file in the bin directory of the CATALINA_HOME directory of your tomcat installation.

Under Ubuntu you should find that directory at /usr/share/tomcat9/bin

If there is no file setenv.sh in this directory, create it as empty file with the rights 755.

Then add to it the following lines:

```
#!/bin/sh
export JAVA_OPTS="-Djdk.xml.xpathExprOpLimit=200"
```

Restart tomcat.

Important hints for Okapi with Tomcat9

Tomcat9 only has rights to write inside of its working directory, which under Ubuntu 18.04 and TomCat9 is

```
/var/lib/tomcat9/work
```

Yet, by default Okapi M37 with Tomcat9 tries to write to

```
/Okapi-Longhorn-files
```

So even if you create a directory at this place with user "tomcat" and rights 777, Okapi will not be able to write files there.

To solve this: Create a config file with the name "okapi-longhorn-configuration.xml" in the tomcat user directory.

Under Ubuntu 18.04 and Tomcat9 this may be the Linux root directory, meaning "/". To find out if this is the case in your Linux distro, you can look up the user directory for the user "tomcat" or "tomcat9" in /etc/passwd.

So for Ubuntu 18.04 and Tomcat9 create a file "okapi-longhorn-configuration.xml" at /okapi-longhorn-configuration.xml with the following contents.

okapi-longhorn-configuration.xml

```
<?xml version="1.0" encoding="utf-8"?>
<okapi-longhorn-configuration version="1">
    <working-directory>/var/lib/tomcat9/work/Okapi-Longhorn-Files</working-directory>
    <use-unique-working-directory>True</use-unique-working-directory>
</okapi-longhorn-configuration>
```

Change the path to "working-directory", if the working directory of your Tomcat9 is located somewhere else.

Ensure that the parent directory of the working directory is writable for the tomcat user, since with the use-unique-working-directory flag = true, Okapi adds the Okapi version to the working directory name.

This is important for updating Okapi in productive envirionments.

Set the rights and user of the new directory (change the path, if your directory is somewhere else) and restart tomcat (change the call to restart tomcat, if this is different in your distro):

```
chown tomcat:tomcat /var/lib/tomcat9/work/Okapi-Longhorn-Files*
chmod 700 /var/lib/tomcat9/work/Okapi-Longhorn-Files*
systemctl restart tomcat9
```

Important hints for Okapi with Tomcat8

Similar as for Tomcat9, also for Tomcat8 the working directory for Okapi needs to be set up manually.

To do this for Ubuntu 18.04, Okapi M37 and Java 1.8 do the following (for other Linux distros or Okapi versions the working directory might differ):

```
mkdir /var/lib/tomcat8/Okapi-Longhorn-Files
chown tomcat8 /var/lib/tomcat8/Okapi-Longhorn-Files
chmod 700 /var/lib/tomcat8/work/Okapi-Longhorn-Files
systemctl restart tomcat8
```

You can find out, if "tomcat8" is really your tomcat user, if you execute the following command, when tomcat is running:

```
ps auxwww | grep -v grep | grep tomcat
```

Trouble shooting, if your Okapi installation is not working

Problems with Okapi working direcory

If you run into problems, please check the tomcat logs for messages like:

> *INFO: The default working directory for Okapi Longhorn will be used, because no other was specified: /usr/share/tomcat7/Okapi-Longhorn-Files*

Ensure that the directory mentioned there exists and is writable for tomcat.

Try to put an Okapi config into the user directory of your tomcat user analogous to what is shown for Okapi installation with Tomcat 9 above.

If this does not help:

Previously we did run into a bug or problem with Okapi, that lead to a dysfunctional Okapi with Tomcat 8 and Okapi M35. So if you run into further problems, either try to set up a newer Okapi version or if you already have the latest Okapi version try to set it up with Tomcat 7 (with JVM 1.7.0_151-b01 Oracle Corporation) or Tomcat 9 (as shown above). The current code block shows the installation of tomcat8. However it seems, that due to a bug in Okapi Longhorn the conversion routines needed by translate5 do not work correctly with Okapi Longhorn running under TomCat 8. Yet with TomCat 7 and the correct Java Version, everything seems to work fine. For more details see attached PDF file..

Translate5 Configuration of the Okapi Plug-In

Config name	Values	Default	Description
runtimeOptions.plugins.Okapi.server	map		Available okapi instances with unique names. Do not change the name after the instance is assigned to a task. {"okapi-longhorn":"http://localhost:8080/okapi-longhorn/"}
runtimeOptions.plugins.Okapi.serverUsed	string		Okapi server used for the a task. All available values are automatically generated out of the runtimeOptions.plugins.Okapi.server config
runtimeOptions.plugins.Okapi.tikal.executable	string		This usually is left empty, because in the default setup of translate5 you do not use Okapi Tikal, but Okapi Longhorn for conversion. The absolute path to the tikal executable, no usable default can be given so is empty and must be configured by the user!
runtimeOptions.worker. editor_Plugins_Okapi_Worker.maxParallelWorkers	integer	3	Max parallel running workers of the Okapi worker

The above configurations must be set in the Zf_configuration table of your translate5 instance.

In addition translate5's Okapi plug-in (which is already part of your installed translate5 instance) must be enabled in your translate5 instance.

To do so, find the right configuration option in your SQL database by executing the following select statement:

Find configuration option 'runtimeOptions.plugins.active'

```
SELECT * FROM `Zf_configuration` WHERE `name` LIKE 'runtimeOptions.plugins.active'
```

Activate translate5's okapi plug-in by adding it to the json array of this config option.

Therefore if the current value column of your config row looks like this

```
["editor_Plugins_Transit_Bootstrap","editor_Plugins_TermTagger_Bootstrap","editor_Plugins_ChangeLog_Init","
editor_Plugins_SpellCheck_Init","editor_Plugins_MatchAnalysis_Init","editor_Plugins_NecTm_Init","
editor_Plugins_DeepL_Init","editor_Plugins_PangeaMt_Init"]
```

Then add the okapi plug-in like this

Activate Okapi plug-in in your configuration

```
UPDATE `Zf_configuration` SET `value` = '[\"editor_Plugins_Okapi_Init\",\"editor_Plugins_Transit_Bootstrap\",\"
editor_Plugins_TermTagger_Bootstrap\",\"editor_Plugins_ChangeLog_Init\",\"editor_Plugins_SpellCheck_Init\",\"
editor_Plugins_MatchAnalysis_Init\"]' WHERE `Zf_configuration`.`name` = 'runtimeOptions.plugins.active';
```

Then update your configuration to make translate5 point to the URL, where your Okapi instance is located, in translate5 interface navigate to http://translate5.local/editor/#preferences/adminConfigGrid and first add your okapi server url(s) (runtimeOptions.plugins.Okapi.server) with unique name and then set this server to be used for all new imports (runtimeOptions.plugins.Okapi.serverUsed)