openTMStermTagger - run as daemon with startup scripts

openTMStermTagger is not part of translate5, but translate5 builds on it to find and highlight terminology.

The script "startServer.sh" starts the openTMStermTagger server as shown in Server environment - configure from scratch.

- Control startup with supervisord (recommended way)
- SysVinit init.d startup script (alternative method - not recommended)
- SystemD start script (alternative method - not recommended)

Control startup with supervisord (recommended way)

Since supervisord is recommended for the WebSocket Server of translate5, it makes sense to control the termtaggers (if they are on the same server) with supervisord too.

There fore use the latest termtagger delivered with translate5 install-and-update script. Add and modify the following example config, thats it.

For the reason why this is the recommended way see OpenTMSTermTagger - recommended setup.

termtagger.ini for supervisord

```
[program:termtagger]
directory               = /path/to/XliffTermTagger
command                 = /path/to/XliffTermTagger/startServer.sh --supervisor http://localhost:900%
(process_num)d
process_name            = termtagger_900%(process_num)s
numprocs                = 2  ;  define here the amount of running termtagger processes
numprocs_start          = 2 ;to start from 9002 on
autostart               = true
autorestart             = true
user                    = tlauria
stopsignal              = INT
stdout_logfile          = /var/log/supervisor/t5-termtagger-900%(process_num)s.log
stdout_logfile_maxbytes = 1MB
stderr_logfile          = /var/log/supervisor/t5-termtagger-900%(process_num)s-error.log
stderr_logfile_maxbytes = 1MB
```

Explanation:

The import part is the numprocs configuration, it defines how many termtagger processes should be started.

The variable process_num is filled with the current number of the process and used in the above example for defining the port (in the line command), the process_name and the logfiles.

Since we want to start with port 9002 here (and not 9000) we have to set numprocs_start to 2

For the termtagger supervisor invocation is at least needed translate5 version 3.4.0

Important: restart supervisord after creating the above configuration file.

```
sudo systemctl restart supervisor
```

After the restart the termtagger(s) server should have been started automatically. To verify this do:

```
sudo supervisorctl status
# this should output then something similar to:
termtagger:termtagger_9001       RUNNING   pid 18505, uptime 0:01:07
```

SysVinit init.d startup script (alternative method - not recommended)

To have openTMStermTagger running on start up, you should start and stop the TermTagger with an init.d script.

Save the following code snippet as file "/etc/init.d/openTMSTermTagger".

Save as /etc/init.d/openTMSTermTagger

```
#!/bin/sh -e
### BEGIN INIT INFO
# Provides:          openTMSTermTagger
# Required-Start:    $syslog
# Required-Stop:     $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: openTMSTermTagger Start daemon at boot time
# Description:       openTMSTermTagger start and stop script, used by Translate5
### END INIT INFO

USER=www-data
APP=openTMSTermTagger
APP_PATH=/var/www/translate5/application/modules/editor/ThirdParty/XliffTermTagger

case "$1" in
  # Start command
  start)
    echo "Starting $APP"
    /bin/su -m $USER -s /bin/bash -c "cd ${APP_PATH} && ${APP_PATH}/startServer.sh &> /dev/null"
    ;;
  # Stop command
  stop)
    echo "Stopping $APP"
    /bin/su -m $USER -s /bin/bash -c "cd ${APP_PATH} && ${APP_PATH}/stopServer.sh &> /dev/null"
    echo "$APP stopped successfully"
    ;;
   # Restart command
   restart)
        $0 stop
        sleep 5
        $0 start
        ;;
  *)
    echo "Usage: /etc/init.d/$APP {start|restart|stop}"
    exit 1
    ;;
esac

exit 0
```

Make the script executable, add it to the default runlevel and start it directly:

```
sudo chmod 755 /etc/init.d/openTMSTermTagger
sudo update-rc.d openTMSTermTagger defaults
sudo /etc/init.d/openTMSTermTagger start
```

Ensure here also that the base termtagger scripts are executable.

```
sudo chmod u+x /var/www/translate5/application/modules/editor/ThirdParty/XliffTermTagger/startServer.sh
sudo chmod u+x /var/www/translate5/application/modules/editor/ThirdParty/XliffTermTagger/stopServer.sh
```

SystemD start script (alternative method - not recommended)

Ensure that the start and stop scripts delivered with translate5 are executable:

```
sudo chmod u+x /var/www/translate5/application/modules/editor/ThirdParty/XliffTermTagger/startServer.sh
sudo chmod u+x /var/www/translate5/application/modules/editor/ThirdParty/XliffTermTagger/stopServer.sh
```

Create an empty serviced unit:

```
sudo touch /etc/systemd/system/termtagger.service
sudo chmod 644 /etc/systemd/system/termtagger.service
```

Copy the following content into the above generated termtagger.service file

```
[Unit]
Description=openTMS TermTagger
After=syslog.target network.target


[Service]
Type=forking
User=www-data
Group=www-data
ExecStart=/var/www/translate5/application/modules/editor/ThirdParty/XliffTermTagger/startServer.sh
ExecStop=/var/www/translate5/application/modules/editor/ThirdParty/XliffTermTagger/stopServer.sh
WorkingDirectory=/var/www/translate5/application/modules/editor/ThirdParty/XliffTermTagger/

[Install]
WantedBy=default.target
```

Ensure that the filepaths used in above ExecStart and ExecStop are pointing to the right place!

Run the following commands to start the termtagger:

```
sudo systemctl daemon-reload
sudo systemctl start termtagger.service

# to start termtagger automatically on boot:
sudo systemctl enable termtagger.service
```