

## rest2site CVS-reStructuredText based CMS

**Author :** Frédéric de Zorzi  
**Contact :** [fredz@\\_nospam\\_pimentech.net](mailto:fredz@_nospam_pimentech.net)  
**Revision :** 1.15  
**Date :** 2008-11-13  
**Copyright :** This document has been placed in the public domain.  
**Tags :** restructuredtext bash

### Contexte d'utilisation

Nous travaillons sur un grand nombre de projets et nous manquons toujours de temps pour communiquer et faire des documentations. Quand ça nous arrive, nous faisons du LaTeX pour le papier, du html pour notre site, et des mails pour des petites infos à partager.

*L'idée est de générer des documentations au format `_reStructuredText` : <http://docutils.sourceforge.net/rst.html>*

### Source

```
#!/bin/bash

RST2HTML=/usr/bin/rst2html.py

usage() {
    echo "rest2site."
    echo "usage : rest2site [ -h for help ] -s <source path> -w <destination path>"
}

cloud() {
    echo
    echo
    echo ".. sidebar:: Tags"
    for tag in $tags ; do
        echo "    \" `\$tag`\""
    done
    echo
    for tag in $tags ; do
        echo "    .. _\$tag:"
        echo "        \$1/\$tag"
    done
    echo
}

while true
do
    case $1 in
```

```

        -d) debug=1 ; set -x ; shift 1 ;;
        -h) usage ; exit 0 ;;
        -s) source=$2 ; shift 2 ;;
        -w) destination=$2 ; shift 2 ;;
        -o) options=$2 ; shift 2 ;;
        *) break ;;
    esac
done

if [ -z "$source" -o -z "$destination" ]
then
    usage
    exit 1
fi

tags=`find $source -name "*.rst" -exec grep ":Tags:" {} \; | cut -d ' ' -f2- | tr

# Index general
mkdir -p $destination/tag
cd $destination/tag
cloud . > index.rst
${RST2HTML} --output-encoding=latin1 index.rst index.html

exit 1

cd $source
for source_file in `find . -name "*.rst" | sed 's/^\.\.\.\//g'`
do
    rst_file=`basename $source_file`
    rst_path=${source_file/$rst_file}
    rel_path=`echo $rst_path | sed -e 's/[^\//]*\//g' -e 's/\.\.\.\//g'`
    html_file=${rst_file/rst}html
    ok_file=${rst_file/rst}ok
    html_destination=$destination/$rst_path$html_file
    cd $source/$rst_path
    link_file=`echo "$rst_path:$html_file" | sed -e 's/\.\.\.\//g' -e 's/^\.\.\.\//g'`
    if [ ! -f $ok_file -o $rst_file -nt $ok_file -o ! -f $html_destination ]
    then
        cloud $rel_path > rest2site.tmp
        cat $rst_file rest2site.tmp | ${RST2HTML} --output-encoding=latin1 $options
        rm rest2site.tmp
        if [ $? -eq 0 ]
        then
            touch $ok_file
            mkdir -p $destination/$rst_path
            cp $rst_file $html_file $destination/$rst_path
            tags=`grep ":Tags:" $rst_file | cut -d ' ' -f 2-`
            for tag in $tags
            do
                mkdir -p $destination/tag/$tag
                ln -sf $html_destination $destination/tag/$tag/$link_file
            done
        fi
        recipients=`grep ":Recipients:" $rst_file | cut -d ' ' -f 2-`
        for recipient in $recipients
        do

```

```

        metasend -b -s "[Garage] Updated $source_file" -m text/html -f $html_file
    done
fi
done

for tag in $tags
do
    cd $destination/tag/$tag
    echo ".. sidebar:: Tags" > index.rst
    echo >> index.rst
    cat ../index.rst >> index.rst
    html_files=`ls *.html|grep -v index.html`
    for html_file in $html_files
    do
        title=`grep '<title>' $html_file | sed 's/<title>\(.*\)</title>/\1/g' ``
        real_file=`find $html_file -printf %l`
        real_file=../../${real_file/$destination}/
        if [ -z "$title" ]
        then
            title=$html_file
        fi
        cat >> index.rst <<EOF
        \`${title}\`_

        .. _$title:
           $real_file

EOF
    done
    ${RST2HTML} --output-encoding=latin1 index.rst index.html
done

# Copier-coller : faire une fonction
cd $destination/tag
rm -f index.rst
tags=`ls -F | grep \/ | sed -e 's\/\$/\/g' `
for tag in $tags
do
    nblinks=`ls ${tag}/*.html|grep -v index.html|wc -l`
    cat >> index.rst <<EOF
    \`${tag}\`_ (${nblinks})

    .. _$tag:
       $tag

EOF
done
${RST2HTML} --output-encoding=latin1 index.rst index.html

```