

Usar la API de XNAT

Uploading Hack

Tengo que separar los DCM por scans. La cosa es que tiene el SeriesNumber en el nombre de archivo asi que deberia ser simple.

```
[osotolongo@detritus mopead]$ ls /nas/corachan/MOPEAD/52JAYALW/ | awk -F"." '{print $4}' | uniq
10001
11001
12001
13
14001
15001
16001
17
18001
19
20001
21
22001
23001
24001
25001
26
27001
28001
5001
6001
7001
8001
9001
```

Voy a crear un tgz por cada serie

```
[osotolongo@detritus mopead]$ for x in `ls /nas/corachan/MOPEAD/52JAYALW/ | awk -F"." '{print $4}' | uniq`; do tar czvf tmp/${x}.tgz /nas/corachan/MOPEAD/52JAYALW/52JAYALW_____.MR.RM_CRANEO_PROT_.$x.*; done
[osotolongo@detritus mopead]$ ls tmp
10001.tgz  12001.tgz  14001.tgz  16001.tgz  18001.tgz  20001.tgz  22001.tgz
24001.tgz  26.tgz     28001.tgz  6001.tgz   8001.tgz
11001.tgz  13.tgz    15001.tgz  17.tgz     19.tgz     21.tgz     23001.tgz
25001.tgz  27001.tgz  5001.tgz   7001.tgz   9001.tgz
```

Pero si subo todo esto voy a tener problemas seguro.A verque sirve de aqui,

```
[osotolongo@detritus mopead]$ for x in `ls /nas/corachan/MOPEAD/52JAYALW/ |
```

```
awk -F"." '{print $4}' | uniq`; do h=$(ls
/nas/corachan/MOPEAD/52JAYALW/52JAYALW_____.MR.RM_CRANEO_PROT_.$x}* | head
-n 1); sd=$(dckey -k "SeriesDescription" $h 2>&1); d=$(dckey -k
"AcquisitionDate" $h 2>&1); if [[ $(echo $d |grep -v Error) ]]; then
echo "$x,$d,$sd";fi; done
1001,20180705,mIP_Images(SW)
1101,20180705,SWI_Images
1201,20180705,ep2d_bold_p2_resting_state
1401,20180705,ep2d_fid_basic_bold_p2_AP
1501,20180705,ep2d_fid_basic_bold_p2_PA
1601,20180705,asl_3d_tra_iso_3.0_highres
1801,20180705,Perfusion_Weighted
2001,20180705,DTIep2d_diff_mddw_48dir_p3_AP
2201,20180705,DTIep2d_diff_mddw_48dir_p3_AP_ADC
2301,20180705,DTIep2d_diff_mddw_48dir_p3_AP_TRACEW
2401,20180705,DTIep2d_diff_mddw_48dir_p3_AP_FA
2501,20180705,DTIep2d_diff_mddw_48dir_p3_AP_ColFA
2701,20180705,DTIep2d_diff_mddw_48dir_p3_AP_TENSOR_B0
2801,20180705,DTIep2d_diff_mddw_4b0_PA
5001,20180705,t1_mprage_sag_p2_iso
6001,20180705,t2_space_dark-fluid_sag_p2_iso
7001,20180705,pd+t2_tse_tra_p2_3mm
8001,20180705,Mag_Images
9001,20180705,Pha_Images
```

Las demas eries dan errores en los tags, no tienen fecha o no tienen *Series Description*. Un horror. En fin que guardo esto en un csv y solo subo estos.

```
[osotolongo@detritus mopead]$ for x in `ls /nas/corachan/MOPEAD/52JAYALW/ |
awk -F"." '{print $4}' | uniq`; do h=$(ls
/nas/corachan/MOPEAD/52JAYALW/52JAYALW_____.MR.RM_CRANEO_PROT_.$x}* | head
-n 1); sd=$(dckey -k "SeriesDescription" $h 2>&1); d=$(dckey -k
"AcquisitionDate" $h 2>&1); if [[ $(echo $d |grep -v Error) ]]; then
echo "$x,$d,$sd";fi; done > tmp/infoups.csv
```

Filtrando la subida

O mejor aun, ya que se los tags (por el protocolo!) que en realidad me interesan, los pongo en un file independiente y solo creo la estructura de estos. Esto es facil porque los tags buenos son igual para cualquier sujeto.

```
[osotolongo@detritus mopead]$ cat tmp/protocol.csv
ep2d_bold_p2_resting_state
ep2d_fid_basic_bold_p2_AP
ep2d_fid_basic_bold_p2_PA
asl_3d_tra_iso_3.0_highres
DTIep2d_diff_mddw_48dir_p3_AP
DTIep2d_diff_mddw_4b0_PA
t1_mprage_sag_p2_iso
```

```
t2_space_dark-fluid_sag_p2_iso
[osotolongo@detritus mopead]$ grep -f tmp/protocol.csv tmp/infoups.csv
12001,ep2d_bold_p2_resting_state
14001,ep2d_fid_basic_bold_p2_AP
15001,ep2d_fid_basic_bold_p2_PA
16001,asl_3d_tra_iso_3.0_highres
20001,DTIep2d_diff_mddw_48dir_p3_AP
28001,DTIep2d_diff_mddw_4b0_PA
5001,t1_mprage_sag_p2_iso
6001,t2_space_dark-fluid_sag_p2_iso
[osotolongo@detritus mopead]$ grep -f tmp/protocol.csv tmp/infoups.csv >
tmp/prot2up.csv
```

Ahora lo que voy a hacer es preparar la estructura para subir todo junto,

```
[osotolongo@detritus mopead]$ for x in `awk -F"," '{print $1}'
tmp/prot2up.csv`; do mkdir tmp/scans/${x}; done
[osotolongo@detritus mopead]$ tree tmp/scans/
tmp/scans/
├── 12001
├── 14001
├── 15001
├── 16001
├── 20001
├── 28001
├── 5001
└── 6001

8 directories, 0 files
[osotolongo@detritus mopead]$ for x in `awk -F"," '{print $1}'
tmp/prot2up.csv`; do cp
/nas/corachan/MOPEAD/52JAYALW/52JAYALW_____.MR.RM_CRANEO_PROT_.${x}.*
tmp/scans/${x}; done
[osotolongo@detritus scans]$ cd tmp/scans
[osotolongo@detritus scans]$ [osotolongo@detritus mopead]$ for x in `awk -
F"," '{print $1}' tmp/prot2up.csv`; do tar czvf tmp/${x}.tar.gz -C
tmp/scans/ ${x}; done
```

Nota: Esto se tiene que hacer mas rapido pasando el paso de copiar!!!

Comandos simples de gestión

Añadir proyecto, (TestAPI)

```
$ curl -X PUT -u osotolongo:kaponko
http://localhost:8088/data/projects/TestAPI
```

Añadir sujeto al proyecto, (Test001)

```
$ curl -X PUT -u osotolongo:kaponko
http://localhost:8088/data/projects/TestAPI/subjects/Test001
XNAT_S00010
```

Subir imagenes. Primero se ha de crear la sesion y el scan,

<https://wiki.xnat.org/display/XAPI/Upload+Image+Session+Files+with+REST+API>

Ahora si, a subir,

```
$ f=$(ls /nas/corachan/MOPEAD/52JAYALW/ | head -n 1); d=$(dckey -k
"AcquisitionDate" /nas/corachan/MOPEAD/52JAYALW/${f} 2>&1); echo $d
20180705
$ f=$(ls /nas/corachan/MOPEAD/52JAYALW/ | head -n 1); pid=$(dckey -k
"PatientID" /nas/corachan/MOPEAD/52JAYALW/${f} 2>&1 | sed 's/^[ \t]*$//');
echo $pid
D18318024

$ curl -X PUT -u osotolongo:kaponko
http://localhost:8088/data/projects/TestAPI/subjects/Test001/experiments/"${
pid}"?xnat:mrSessionData/date="${d}"
$ for x in `awk -F"," '{print $1}' tmp/prot2up.csv`; do curl -X PUT -u
osotolongo:kaponko
http://localhost:8088/data/projects/TestAPI/subjects/Test001/experiments/"${
pid}"/scans/"${x}"?xsiType=xnat:mrScanData; done
$ for l in `cat
tmp/prot2up.csv`; do x=$(echo ${l} |awk -F"," '{print $1}'); tag=$(echo ${l}
|awk -F"," '{print $2}'); curl -X PUT -u osotolongo:kaponko
"http://localhost:8088/data/projects/TestAPI/subjects/Test001/experiments/${
pid}/scans/${x}?xsiType=xnat:mrScanData&xnat:mrScanData/series_description=${
tag}&xnat:imageScanData/quality=usable&xnat:imageScanData/type=${tag}";
done
$ for x in `awk -F"," '{print $1}' tmp/prot2up.csv`; do curl -X PUT -u
osotolongo:kaponko
http://localhost:8088/data/projects/TestAPI/subjects/Test001/experiments/"${
pid}"/scans/"${x}"/resources/DICOM/?format=DICOM; done
$ for x in `awk -F"," '{print $1}' tmp/prot2up.csv`; do curl -X PUT -u
osotolongo:kaponko
http://localhost:8088/data/projects/TestAPI/subjects/Test001/experiments/"${
pid}"/scans/"${x}"/resources/DICOM/files?extract=true -F
"file.tar.gz=@tmp/"${x} ".tar.gz"; done
```

PROJECT: TestAPI > SUBJECT: Test001 > D18318024

MR Session: D18318024

Details	Projects	Actions
Accession #: XNAT_E00032 Date Added: 10/23/2019 10:37:02 (osotolongo) Date: 07/05/2018	Subject: Test001 Gender: Handedness: Age: --	Edit View Download Email Manage Files View Images Delete

Scans

Bulk Actions:		Download					
<input type="checkbox"/>	Scan	Type	Series Desc	Usability	Files	Note	
<input type="checkbox"/>	5001	t1_mprage_sag_p2_iso	t1_mprage_sag_p2_iso	usable	33.9 MB in 176 files		
<input type="checkbox"/>	6001	t2_space_dark-fluid_sag_p2_iso	t2_space_dark-fluid_sag_p2_iso	usable	38.7 MB in 192 files		
<input type="checkbox"/>	12001	ep2d_bold_p2_resting_s_tate	ep2d_bold_p2_resting_s_tate	usable	809.7 MB in 7600 files		
<input type="checkbox"/>	14001	ep2d_fid_basic_bold_p2_AP	ep2d_fid_basic_bold_p2_AP	usable	3.8 MB in 38 files		
<input type="checkbox"/>	15001	ep2d_fid_basic_bold_p2_PA	ep2d_fid_basic_bold_p2_PA	usable	3.9 MB in 38 files		
<input type="checkbox"/>	16001	asl_3d_tra_iso_3.0_high_res	asl_3d_tra_iso_3.0_high_res	usable	36.9 MB in 320 files		
<input type="checkbox"/>	20001	DT1ep2d_diff_mddw_48_dir_p3_AP	DT1ep2d_diff_mddw_48_dir_p3_AP	usable	513.5 MB in 3456 files		
<input type="checkbox"/>	28001	DT1ep2d_diff_mddw_4b_0_PA	DT1ep2d_diff_mddw_4b_0_PA	usable	8.8 MB in 64 files		

Total: 1.4 GB in 11884 files

Nota: Hacer scripting de esto y ya tengo el primer paso de la *pipeline v0.4*, **upload_subject**

El código, en forma de script, sería así:

```
#!/bin/bash

ARGS=( $@ )
if [ ${#ARGS[@]} -lt 3 ] ; then
  echo "Usage:"
  echo " $0 <DICOM folder> <Xnat project ID> <Xnat subject>"
  exit 0
fi

DIR="${ARGS[0]}"
PROY="${ARGS[1]}"
SUBJECT="${ARGS[2]}"

#configuración
XNAT=http://detritus.fundacioace.com:8080
USER=usuario:contrasenna

#temporary files
TMP_PRO=$(mktemp)
TMP_VAL=$(mktemp)
```

```

TMP_SCS=$(mktemp -d)

#Tipos de estudios
cat > $TMP_PRO <<.EOF
ep2d_bold_p2_resting_state
ep2d_fid_basic_bold_p2_AP
ep2d_fid_basic_bold_p2_PA
asl_3d_tra_iso_3.0_highres
DTIep2d_diff_mddw_48dir_p3_AP
DTIep2d_diff_mddw_4b0_PA
t1_mprage_sag_p2_iso
t2_space_dark-fluid_sag_p2_iso
.EOF

#Archivos de la misma serie ($4 único), con fechas y descripciones,
filtrados por tipo de estudio
for x in $(ls "$DIR" | awk -F"." '{print $4}' | uniq); do
    h=$(ls "$DIR" | grep '^([A-Z0-9_]\+\.([A-Z0-9_]\+\.([A-Z0-9_]\+\. '$x'\. ' |
head -n 1)
    d=$(dckey -k "AcquisitionDate" "$DIR/$h" 2>&1 | grep -v Error) &&
sd=$(dckey -k "SeriesDescription" "$DIR/$h" 2>&1 | grep -v Error) && echo
"$x,$d,$sd"
done | grep -f $TMP_PRO > $TMP_VAL
# salida: series de archivos válidas

#Guardar los archivos de cada serie en un TAR.GZ
for x in $(awk -F"," '{print $1}' $TMP_VAL); do
    #NOTA: usamos el truco de conversión de nombre del tar para no tener que
copiar datos
    tar czvf $TMP_SCS/$x.tar.gz --transform='s/^[^$DIR]\\/'$x'\\/' -C "$DIR"
$(ls "$DIR" | grep '^([A-Z0-9_]\+\.([A-Z0-9_]\+\.([A-Z0-9_]\+\. '$x'\. '))
done

#Todos los archivos en el directorio son de la misma fecha y paciente
DATE=$(dckey -k "AcquisitionDate" "$DIR"/$(head -n 1 $TMP_VAL | awk -F"," '{print $4}') 2>&1)
PTID=$(dckey -k "PatientID" "$DIR"/$(head -n 1 $TMP_VAL | awk -F"," '{print $4}') 2>&1 | sed 's/[ \t]$/g')

#Guardar los archivos de cada serie en un TAR.GZ
for x in $(awk -F"," '{print $1}' $TMP_VAL); do
    #NOTA: usamos el truco de conversión de nombre del tar para no tener que
copiar datos
    tar czvf $TMP_SCS/$x.tar.gz --transform='s/^[^$DIR]\\/'$x'\\/' $(ls
"$DIR"/* | grep '\([A-Z0-9_]\+\.([A-Z0-9_]\+\.([A-Z0-9_]\+\. '$x'\. '))
done

#Crear el proyecto, el paciente y el experimento
curl -X PUT -u $USER "$XNAT/data/projects/$PROY"
curl -X PUT -u $USER "$XNAT/data/projects/$PROY/subjects/$SUBJECT"
curl -X PUT -u $USER

```

```
"$XNAT/data/projects/$PROY/subjects/$SUBJECT/experiments/$PTID?xnat:mrSessionData/date=$DATE"

#Cargar los datos de las series (para ahorrar espacio en disco, se podría integrar aquí la creación de los tar.gz)
cat $TMP_VAL | while read l ; do
  x=$(echo "$l" | awk -F"," '{print $1}')
  TAG=$(echo "$l" | awk -F"," '{print $2}')
  #crear la serie de tipo mrScanData
  curl -X PUT -u $USER
"$XNAT/data/projects/$PROY/subjects/$SUBJECT/experiments/$PTID/scans/$x?xsiType=xnat:mrScanData"
  #poner la descripción de la serie (el TAG)
  curl -X PUT -u $USER
"$XNAT/data/projects/$PROY/subjects/$SUBJECT/experiments/$PTID/scans/$x?xsiType=xnat:mrScanData&xnat:mrScanData/series_description=$TAG&xnat:imageScanData/quality=usable&xnat:imageScanData/type=$TAG"
  #definir un recurso DICOM
  curl -X PUT -u $USER
"$XNAT/data/projects/$PROY/subjects/$SUBJECT/experiments/$PTID/scans/$x/resources/DICOM/?format=DICOM"
  #cargar los DICOM en los TAR.GZ (esto, parece que no funciona en la parte del curl)
  curl -X PUT -u $USER
"$XNAT/data/projects/$PROY/subjects/$SUBJECT/experiments/$PTID/scans/$x/resources/DICOM/files?extract=true" -F "file.tar.gz=@$TMP_SCS/$x.tar.gz"
done

#Dejar todo limpio
rm -rf $TMP_PRO $TMP_VAL $TMP_SCS
```

From:
<https://mail.fundacioace.com/wiki/> - Detritus Wiki

Permanent link:
https://mail.fundacioace.com/wiki/doku.php?id=neuroimagen:xnat_api

Last update: **2020/08/04 10:58**

