

XNAT API Client

API de Xnat

Xnat implementa servicios web basados en una API basada en la especificación [REST](#) (representational state transfer). Esta API es muy sencilla: llamadas GET, POST, PUT, DELETE del protocolo HTTP con URLs construidos de forma sencilla, p.ej.

GET /data/projects/**MyProject**/subjects/**SubjectNo1**/experiments/**FirstExperiment**

Aquí las palabras en negrilla indican los valores específicos del objeto al que nos queremos referir (el tipo de objeto es el que los antecede, escrito en plural).

La API de Xnat es redundante, en el sentido de que existen diferentes formas de realizar una misma operación; esto está relacionado con el modelo de datos de Xnat, que asigna un identificador único a cada sujeto de manera que puedan pertenecer a varios proyectos, de modo que algunas operaciones se pueden realizar refiriéndose al sujeto dentro del proyecto o al sujeto como objeto del sistema. Y lo mismo sucede para los experimentos.

Otros temas a desarrollar aquí como contexto:

- scan: pertenecen al experimento (conjuntos de imágenes)
- recursos: pertenecen al experimento y pueden ser de distintos tipos (e.g. DICOM, NIFTY, etc.)
- archivo y prearchivo: las imágenes se pueden prearchivar, visualizar y, posteriormente, enviar al archivo.
- ejecución de pipelines: son las operaciones de procesado que se ejecutan al archivar las imágenes.

Instalación de xnatapic

Descomprimir el archivo comprimido en /nas/usr/local.

En /nas/usr/local/bin se copiará el script ejecutable xnatapic que es al que se llamará para ejecutar los comandos. El resto de los scripts y otros archivos se instalarán como *apps* bajo /nas/usr/local/share/xnatapic/.

Configuración de usuario

El usuario debe crear en su *home* el directorio *oculto* ~/ .xnatapic.

En este directorio debe crear un archivo de configuración particular xnat.conf con el siguiente contenido:

```
HOST=http://detritus.fundacioace.com:8088
USER=usuario
```

```
PASSWORD=contrasenna
```

Bajo este directorio ~/ .xnatapic/ se pueden crear otros scripts ayudantes que serán llamados desde xnatapic (véase el [Apéndice: estructura de los scripts](#)).

Nota: en ~/ .xnatapic/xnat.conf se pueden añadir otras variables u otras instrucciones que se necesiten en los scripts del usuario. Por ejemplo:

```
USERNAME=SotoLongoGrau,Oscar
```

(que se usa para indicar el nombre del investigador principal de un proyecto)

Tests de uso

Las llamadas empiezan siempre con xnatapic seguido del nombre de la operación a realizar o de parámetros (de la forma -nombre) y valores que los scripts usan como parámetros de su ejecución.

Ayuda de xnatapic

Todos los scripts tienen una ayuda básica que muestra la sintaxis de su llamada. Por ejemplo, la ayuda de xnatapic muestra todas las operaciones que se pueden hacer:

```
$ xnatapic --help
xnatapic - call XNat procedures from the command line
* create_project - create a Xnat project
* create_subject - create a subject in a project
* delete_project - delete a project owned by the user
* delete_subject - delete a subject in a project
* list_projects - list projects owned by the user
* list_subjects - list subjects in a project
* upload_dicom - upload a DICOM folder to Xnat repository
```

(en el futuro se añadirán más)

Sobre cada una de las operaciones se puede consultar la ayuda, por ejemplo:

```
$ xnatapic create_project --help
create_project - create a Xnat project
--help: show this help
--project_id <project id> [mandatory]
--name <project name> [defaults to <project id>]
--desc <project description> [defaults to <project id>]
--PI <apellidos,nombre>
--keywords <key,key,key>
```

Creación de proyectos, sujetos y experimentos

Como se ve en la ayuda anterior, la forma más simple de crear un proyecto es:

```
xnatapic create_project --project_id MyProject
```

Éste será un proyecto vacío (sin sujetos, ni estudios) y sin más propiedad que el nombre. Las demás opciones de línea de comandos permiten asignar un nombre alternativo (-name), una descripción más extensa (-desc), el nombre del investigador principal (-PI) y las palabras clave del proyecto (-keywords). Estas opciones se pueden indicar inicialmente o llamando a `xnatapic create_project` una vez creado el proyecto; lo mismo vale para cambiar estos atributos del proyecto:

```
xnatapic create_project --project_id MyProject --desc "Una prueba de
proyecto en Xnat"
xnatapic create_project --project_id MyProject --PI SotolonguGrau,Oscar
```

Nota: Usando una instrucción no documentada de Xnat, también se puede cambiar el acceso al proyecto, para que sea *protegido*, *privado* o *público*.

```
xnatapic create_project --project_id MyProject --access public
```

Para definir sujetos, se usa la operación `create_subject`. Se debe indicar el proyecto (-project_id) dentro del que se crean:

```
xnatapic create_subject --project_id MyProject --subject_id Subject001
```

Los sujetos se crean con el ID indicado antes, pero también se les puede asignar una etiqueta más fácilmente identificable. Una vez asignada, será indistinto referirse a ellos por su ID de sujeto o por esa etiqueta:

```
xnatapic create_subject --project_id MyProject --subject_id Subject001 --
label "FirstSubject"
xnatapic create_subject --project_id MyProject --subject_id FirstSubject --
gender M --handedness R
```

Listado de proyectos, sujetos y experimentos

```
xnatapic list_projects --name --desc
```

```
xnatapic list_subjects --project_id MyProject --label --gender --handedness
--age
```

Borrado de proyectos, sujetos y experimentos

```
xnatapic delete_project --project_id MyProject
```

```
xnatapic delete_subject --project_id MyProject --subject_id Subject001
```

Carga imágenes DICOM

Las imágenes DICOM pueden encontrarse separadas por series o todas ellas en un mismo directorio (con las “series mezcladas”). Las series separadas en directorios, se pueden subir una a una con

```
xnatapic upload_dicom --project_id MyProject
/nas/home/daniel/data/DC182211/6001/
xnatapic upload_dicom --project_id MyProject
/nas/home/daniel/data/DC182211/10001/
```

Si las series no están separadas, sino que están mezcladas en un directorio:

```
xnatapic upload_dicom --project_id MyProject --mixed-series
/nas/home/daniel/data/52JAYALW/
```

Nota: Aunque en teoría Xnat debería ser capaz de separar las series cargándolas desde un mismo directorio (lo hace cuando se cargan desde la GUI, subiendo un ZIP), la API da error.

Listado y ejecución de pipelines

Para consultar los pipelines que han sido añadidos a un proyecto, se puede ejecutar:

```
xnatapic list_pipelines --project_id MyProject --path
```

La opción `--path` pide que se muestre, además del nombre del pipeline, la ruta al XML que lo define.

Algo más interesante puede ser saber qué pipelines se pueden añadir a un proyecto, o sea, cuáles ha cargado el administrador de Xnat. Esto se hace con la opción `--available`:

```
xnatapic list_pipelines --project_id MyProject --available --path
```

Un pipeline que se haya añadido a un proyecto sin indicar que se lanzará cuando se archiven las imágenes, se podrá ejecutar posteriormente sobre los experimentos (todos o sólo alguno) de ese proyecto:

```
xnatapic run_pipeline --project_id MyProject --pipeline CleanMRSession --
dcmTags default
```

El argumento `--dcmTags` en este caso es un argumento del pipeline `CleanMRSession` (indica los tags que deben estar presentes para que la sesión sea considerada válida).

Apéndice: extensión de "xnatapic" más allá de lo que permite la API de Xnat

Algunas operaciones que podría ser interesante realizar desde el cliente `xnatapic` no están disponibles a través de la API de Xnat. Por ejemplo, declarar (como administrador) los pipelines disponibles en Xnat, o definir (como propietario de un proyecto) los pipelines que se lanzarán desde un proyecto, o borrar los pipelines de un proyecto. Sin embargo, esto se puede hacer desde la GUI.

Así que una opción es *imitar el comportamiento de la GUI* o, como en el caso del borrado, recurrir a la API no documentada.

Se muestran los ejemplos siguientes:

```
xnatapic define_pipeline --path
/nas/data/xnat/pipeline/catalog/CleanMRSession/CleanMRSession.xml
```

```
xnatapic append_pipeline --project_id MyProject --pipeline CleanMRSession
```

```
xnatapic delete_pipeline --project_id MyProject --pipeline CleanMRSession
```

Nota: se recomienda colocar estos scripts (), que usan métodos no documentados, en el directorio `.xnatapic` del usuario que los quiera usar; en versiones anteriores, alguno de estos scripts dejaba inestable la base de datos de Xnat.

Tareas pendientes

1. Añadir/listar/borrar scans de un experimento
2. Descargar scans

Apéndice: estructura de los scripts

1. Variables globales
2. Variables locales
3. Interpretación de las opciones de línea de comandos
4. Comprobación de la lógica de las opciones
5. Ejecución de la instrucción llamando a la API con curl

Ejemplo de script:

```
#!/bin/bash

ARGS=( "$@" )
HELP="create_project - create a Xnat project"

#local variables
PROJ_ID=""
PROJ_NAME=""
PROJ_DESC=""
PROJ_PI=""
PROJ_KEYS=""

#global variables
[ -s "$XNATAPIC_APPS/xnat.conf" ] && . "$XNATAPIC_APPS/xnat.conf"
[ -s "$XNATAPIC_HOME_APPS/xnat.conf" ] && . "$XNATAPIC_HOME_APPS/xnat.conf"
```

```
#parse arguments
for ((n=0; n<${#ARGS[@]}; n++)) ; do
  case "${ARGS[$n]}" in
    --help-short)
      echo "$HELP"
      exit 0
      ;;
    --help)
      echo "$HELP"
      cat <<.EOF
--help: show this help
--project_id <project id> [mandatory]
--name <project name> [defaults to <project id>]
--desc <project description> [defaults to <project id>]
--PI <apellidos,nombre>
--keywords <key,key,key>
.EOF
      exit 0
      ;;
    --project_id)
      let n=n+1
      PROJ_ID="${ARGS[$n]}"
      ;;
    --name)
      let n=n+1
      PROJ_NAME="$(echo "${ARGS[$n]}" | sed 's/[ \n\r\t]\+/\+g')"
      ;;
    --desc)
      let n=n+1
      PROJ_DESC="$(echo "${ARGS[$n]}" | sed 's/[ \n\r\t]\+/\+g')"
      ;;
    --PI)
      let n=n+1
      PROJ_PI="$(echo "${ARGS[$n]}" | sed 's/[ \n\r\t]\+/\+g')"
      ;;
    --keywords)
      let n=n+1
      PROJ_KEYS="$(echo "${ARGS[$n]}" | sed 's/[ \n\r\t]\+/\+g')"
      ;;
    -*)
      echo "Warning: ignoring option or command ${ARGS[$n]}" >&2
      ;;
    *)
      echo "Warning: ignoring option or command ${ARGS[$n]}" >&2
      ;;
  esac
done

#checks
if [ -z "$PROJ_ID" ]; then
  echo "Error: a project ID is required" >&2
```

```

    exit 1
fi
#defaults
[ -z "$PROJ_NAME" ] && PROJ_NAME="$PROJ_ID"
if [ -z "$PROJ_PI" ] && ! [ -z "$USERNAME" ] ; then
    PROJ_PI="$USERNAME"
fi
if ! [ -z "$PROJ_DESC" ] ; then
    PROJ_DESC="&description=$PROJ_DESC"
fi
if ! [ -z "$PROJ_PI" ] ; then
    PROJ_PI_L="$( echo "$PROJ_PI" | sed 's/ *,.*$//' | sed 's/[
\t\r]\+/+/g' )"
    PROJ_PI_F="$( echo "$PROJ_PI" | sed 's/^.*, *//' | sed 's/[
\t\r]\+/+/g' )"
    PROJ_PI="&pi_firstname=$PROJ_PI_F&pi_lastname=$PROJ_PI_L"
fi
if ! [ -z "$PROJ_KEYS" ] ; then
    PROJ_KEYS="&keywords=$PROJ_KEYS"
fi

#run
echo curl -f -X PUT -u "$USER:$PASSWORD"
"$HOST/data/projects/$PROJ_ID?name=$PROJ_NAME$PROJ_DESC$PROJ_PI$PROJ_KEYS"
if ! curl -f -X PUT -u "$USER:$PASSWORD"
"$HOST/data/projects/$PROJ_ID?name=$PROJ_NAME&description=$PROJ_DESC$PROJ_PI
$PROJ_KEYS" ; then
    echo "Error: server reported an error" >&2
    exit 1
fi

```

Apéndice: ayuda completa de xnatapic

La ayuda completa que se muestra en el cuadro inferior se ha generado con la salida de las instrucciones:

```

xnatapic --help
echo
for a in $(xnatapic --help | grep '^s*\*' | sed 's/^\s*\*//' | sed 's/\s-
\s.*$//' | grep -v get_) ; do
    xnatapic $a --help
    echo
done

```

Ayuda completa de "xnatapic" (16/12/2020):

```
* append_pipeline - append a pipeline to a project (UNDOCUMENTED)
```

- * create_experiment - create an experiment and set its attributes
- * create_project - create a Xnat project
- * create_subject - create a subject in a project
- * define_pipeline - define a pipeline in Xnat from its path (UNDOCUMENTED)
- * delete_experiment - delete an experiment
- * delete_pipeline - delete a pipeline from a project (UNDOCUMENTED)
- * delete_project - delete a project owned by the user
- * delete_subject - delete a subject in a project
- * list_experiments - list experiments associated with a project
- * list_pipelines - list pipelines in a project
- * list_projects - list projects owned by the user
- * list_subjects - list subjects in a project
- * run_pipeline - run a project pipeline on an experiment (or on all the experiments) (UNDOCUMENTED)
- * undefine_pipeline - remove a pipeline definition from Xnat (UNDOCUMENTED)
- * upload_dicom - upload a DICOM folder to Xnat repository

append_pipeline - append a pipeline to a project (UNDOCUMENTED)
--help: show this help
--project_id <project id> [mandatory]
--pipeline <pipeline name> [mandatory]

WARNING: this command does not use Xnat API.

create_experiment - create an experiment and set its attributes
--help: show this help
--project_id <project id> [mandatory]
--subject_id <subject ID> [mandatory]
--experiment_id <experiment ID> [mandatory]
--type <MR,PET> [mandatory]
--date <dd/mm/yyyy or yyyymmdd>
--label <label>

create_project - create a Xnat project
--help: show this help
--project_id <project id> [mandatory]
--name <project name> [defaults to <project id>]
--desc <project description> [defaults to <project id>]
--PI <apellidos,nombre>
--keywords <key,key,key>
--access <private,public,protected> [defaults to protected]

create_subject - create a subject in a project
--help: show this help
--project_id <project id> [mandatory]
--subject_id <subject ID> [mandatory either this or a label]
--label <label>
--gender <male,female>


```
--handedness <right,left,ambidextrous,unknown>  
--age <YY>
```

```
define_pipeline - define a pipeline in Xnat from its path (UNDOCUMENTED)  
--help: show this help  
--path <pipeline path> [mandatory]
```

WARNING: this command does not use Xnat API.

```
delete_experiment - delete an experiment  
--help: show this help  
--project_id <project id> [mandatory]  
--subject_id <subject ID> [mandatory]  
--experiment_id <experiment ID> [mandatory]
```

```
delete_pipeline - delete a pipeline from a project (UNDOCUMENTED)  
--help: show this help  
--project_id <project id> [mandatory]  
--pipeline <pipeline name> [mandatory]
```

WARNING: this command uses an undocumented API call.

```
delete_project - delete a project owned by the user  
--help: show this help  
--project_id: project to be deleted
```

```
delete_subject - delete a subject in a project  
--help: show this help  
--project_id: subject's project  
--subject_id: subject's ID  
--label: subject's label
```

```
list_experiments - list experiments associated with a project  
--help: show this help  
--project_id <project id>  
--subject_id <subject ID>  
--experiment_id <experiment ID>  
--date-range <dd/mm/yyyy[-dd/mm/yyyy]>  
--modality <modality>  
--date  
--date-insertion  
--label  
--project-name  
--type
```

```
list_pipelines - list pipelines in a project
--help: show this help
--project_id <project id>: project
--available: show available (i.e. not active) pipelines
--path: show path
--description: show description
--applies-to: show what data the pipeline is applied to
```

```
list_projects - list projects owned by the user
--help: show this help
--project_id: show only this project
--name: show project names
--desc: show project descriptions
```

```
list_subjects - list subjects in a project
--help: show this help
--project_id: project
--subject_id: show only this subject
--label: show subject label
--gender: show gender
--handedness: show handedness
--age: show subject's age
```

```
run_pipeline - run a project pipeline on an experiment (or on all the
experiments) (UNDOCUMENTED)
--help: show this help
--project_id <project id> [mandatory]
--pipeline <pipeline name> [mandatory]
--experiment_id <experiment ID or ALL (default)>
--[pipeline argument] <argument value>
```

WARNING: this command uses undocumented Xnat API.

```
undefine_pipeline - remove a pipeline definition from Xnat (UNDOCUMENTED)
--help: show this help
--path <pipeline path> [mandatory]
```

WARNING: this command does not use Xnat API.

```
upload_dicom - upload a DICOM folder to Xnat repository
--help: show this help
--project_id <project id>: the project the images belong to [mandatory]
--subject_id <subject ID>: the subjects of the images [optional]
--session <session ID>: the session the images were acquired [IGNORED]
--experiment_id <experiment ID>: the ID of the experiment [optional]
--mixed-series: the folder contains different series
```

```
--pipelines: run pipelines (i.e. upload to prearchive, then archive and launch pipelines)
--append: append uploaded data to already existing in the experiment
--progress: show upload progress
--zip: use a ZIP file containing the DICOM series [mandatory if no dicom_dir folder is provided]
<dicom_dir folder>: list of folders containing DICOM images [mandatory if no zip file is provided]
```

From:

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

Permanent link:

https://mail.fundacioace.com/wiki/doku.php?id=neuroimagen:xnat_apiclient

Last update: **2020/12/16 10:19**

