

# OSUG-DOI (public)

<http://doi.osug.fr/>

## Principe



Roles:

- Datacite: assure la résolution du DOI et possède les informations pour chaque DOI: méta données (xml) + URL associée vers doi.osug.fr (landing page ou ressource)
- OSUG-DC: le serveur doi.osug.fr héberge les landing pages (en local) ou réalise une redirection vers un autre serveur (spip...) dans le périmètre de l'OSUG
- les landing pages doivent être pérennes et contenir les instructions pour accéder aux données / services

## Développement

Source code: <https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi>

TODO:

- DOC:
  1. écrire la Procédure pour demander des DOIs (intro, approche, méta-données), initier un projet (templates, config...)
  2. créer template CSV (with all possible key/values + comments)
- DEV:
  1. supervision pour detecter les URLs invalides (URL 404) et basculer sur les landing page locales
- PROD:
  1. recréer vm doi.osug.fr (ansible) et déploiement application automatisé

Architecture générale:

## Génération Landing Pages (datacite XML to HTML)

A partir d'un document XML correspondant aux métadonnées du DOI (datamodel datacite), une transformation XSLT génère une page HTML statique (bootstrap + CSS) qui présente simplement les méta données et les instructions d'accès aux données / service.

## Outil de génération des métadonnées DOI (datacite XML)

Cas d'utilisation = AMMA-CATCH (voir <http://bd.amma-catch.org/>) ou OHMCV (fiches Sedoo):

Certaines meta-données sont déjà renseignées pour chaque jeu sur le portail Web d'accès aux données AMMA-CATCH et sont récupérables en utilisant le webservice CSW méthode getRecords (format XML ISO 19139):

<http://bd.amma-catch.org/amma-catchWS2/WS/csw/default?service=CSW&request=GetRecords&version=2.0.2&typenames=csw:Record&resulttype=results&maxrecords=100&elementsetname=summary&outputschema=http://www.isotc211.org/2005/gmd>

Il faut ensuite les compléter pour produire un fichier XML complet (dataCite) pour chaque jeu de données.

Solution: pour rester indépendant du format datacite (XML), des fichiers CSV (clé / valeur) sont utilisés pour définir tout ou partie des méta-données.

## Configuration d'un projet

Un projet dans l'outil OSUG-DOI correspond à un ensemble de DOIs au préfixe commun, comme un service d'observation, un data paper.

Toute la configuration du service OSUG-DOI avec tous les projets est archivée dans git (forge gricad): <https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi/tree/master/conf>

Pour compléter les metadonnées obtenues du webservice CSW, des templates (global, par pays, par jeu de données) au format CSV sont utilisés:

<https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi/tree/master/conf/AMMA-CATCH/templates>

Pour indiquer l'URL d'une landing page externe (spip...), le fichier doi\_url\_landing\_page.csv ( table [DOI suffix :: URL (absolue) ] ) est utilisé:

[https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi/blob/master/conf/AMMA-CATCH/doi\\_url\\_landing\\_page.csv](https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi/blob/master/conf/AMMA-CATCH/doi_url_landing_page.csv)

Pour donner l'URL associée réellement au DOI (accès aux données), le fichier doi\_url\_data\_access.csv ( table [DOI suffix :: URL (absolue) ] ) est utilisé:

[https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi/blob/master/conf/OHMCV/doi\\_url\\_data\\_access.csv](https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi/blob/master/conf/OHMCV/doi_url_data_access.csv) Si le DOI ou ce fichier est absent, la valeur par défaut définie dans le fichier project.properties (dataAccessUrl=...) est utilisée.

Pour définir les instructions d'accès aux données, il faut fournir le fragment HTML:

[https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi/blob/master/conf/AMMA-CATCH/access\\_instruction.html](https://gricad-gitlab.univ-grenoble-alpes.fr/OSUG/DC/osug-doi/blob/master/conf/AMMA-CATCH/access_instruction.html)



## Chaine de fusion des informations (CSW + templates + URLs)

Préparation CSW (AMMA-CATCH / OHMCV) (bash)

1. GetRecords → renvoie autant d'éléments <gmd:MD\_Metadata/> qu'il y a de jeux de données
2. Conversion au format CSV

## Séquence de traitement:

1. Pour chaque jeu de données partiel (/inputs), fusionner avec les templates (ajout, pas de suppression):
  1. template global: template\_all.csv
  2. template du pays: correspondant au pays indiqué par geoLocationPlace (geoLocationPlace;Benin)
  3. template spécifique au jeu de données: correspondant au suffixe du DOI (identifiant:DOI;10.5072/AMMA-CATCH.CL.Run\_O)
2. Pour chaque jeu de données:
  1. vérifier les related identifiers (présent dans la BDD OSUG-DOI ?) pour tester les références invalides
  2. convertir au format XML datacite
  3. validation du fichier XML avec le schéma datacite
  4. vérifier les URLs (landing page externe et l'URL associée au DOI)
  5. générer la landing page associée (html) avec l'accès aux données (fragment html + URL)
  6. tester la publication du DOI (metadata + URL OSUG-DOI) avec le préfixe de test: STAGING
  7. sauvegarder l'état du DOI en BDD (statut, validation, URL ...)
3. ré-générer les pages d'index, de rapport d'erreur (STAGING) et les redirections(/r) (internes + landing pages externes)



## Notes:

1. les identifiants DOI présents dans la configuration (CSV) ne doivent contenir que le préfixe de test 10.5072 (publication ultérieure)
2. les méta-données en doublons sont ignorées (même clé / valeur) mais attention si les informations ne sont pas disjointes (clés présentes dans plusieurs templates)
3. attention à l'encodage des fichiers CSV (UTF-8 sous linux et iso-8859-1 sous windows) !
4. il est possible d'utiliser des fichiers Excel pour saisir les templates (qui seront convertis correctement en CSV)

Pendant la phase de préparation ou de mise à jour des DOIs, il faut éditer les fichiers CSV, relancer la chaîne de traitement et vérifier les résultats en STAGING: <http://doi.osug.fr/staging/index.html>

## Publication DOI (datacite)

Lorsqu'un ou plusieurs DOIs sont valides et vérifiés, il est possible de publier le DOI sur datacite (attribution) ce qui le rend pérenne.

## Séquence de publication d'un DOI:

1. vérifications
2. copier les méta-données XML (datacite)
3. ré-écrire les identifiants DOI (identifier et relatedIdentifier) pour remplacer le préfixe de test 10.5072 en préfixe OSUG
4. générer la landing page en PUBLIC (URL publiée chez datacite)
5. publier le DOI (metadata + URL OSUG-DOI) avec le préfixe OSUG
6. mettre à jour l'état du DOI en BDD
7. ré-générer les pages d'index et les redirections(/r) (internes + landing pages externes)

From:

<https://wiki-uar.osug.fr/> - **Wiki de l'UAR OSUG**

Permanent link:

[https://wiki-uar.osug.fr/doku.php?id=osug-dc:2-suivi\\_projets:doi:osug-doi](https://wiki-uar.osug.fr/doku.php?id=osug-dc:2-suivi_projets:doi:osug-doi)

Last update: **2019/01/22 13:54**

