



Release notes

Version 6.1.0

Juin 2023



Nouveautés back-office

Évolutions fonctionnelles

Collecte / Entrée :

- Module de collecte : ajout de règles pour le classement automatisé des archives collectées sur la base de métadonnées portées par les archives et d'un paramétrage du projet de versement

Recherche et accès aux archives :

- Export DIP et transfert d'archives: calcul et enregistrement de l'empreinte du DIP de communication ou du SIP de transfert dans le journal des opérations afin de permettre un contrôle d'intégrité par l'utilisateur en réception du paquet transmis.

Gestion des données :

- Normalisation et contrôle de l'enregistrement des métadonnées techniques « hors SEDA » permettant l'usage de vocabulaire techniques externes

Référentiel des formats :

- Montée de version du référentiel PRONOM (version 109) et du logiciel d'analyse d'identification de format SIEGFRIED

Évolutions techniques

- Montée de version de cots :
 - MongoDB vers latest bugfix (5.0.13 -> 5.0.14).
 - Consul vers latest bugfix (1.12.4 -> 1.12.9).
 - Elastic components vers latest bugfix (7.16.3 -> 7.17.8).
 - curator vers latest version (5.8.1 -> 5.8.4).
- Montée de version d'extras :
 - grafana vers latest version (8.4.1 -> 9.3.6).
 - alertmanager vers latest version (0.20.0 -> 0.25.0).
 - prometheus vers latest version (2.19.0 -> 2.42.0).
- Montée de version d'exporters Prometheus :
 - node_exporter vers latest version (1.0.0 -> 1.5.0).
 - elastic_exporter vers latest version (1.3.0 -> 1.5.0).
 - consul_exporter vers latest version (0.8.0 -> 0.9.0).
- Supervision de la reconstruction via Prometheus
- OS :
 - Debian : compatibilité du déploiement à la fois sur debian 10 et debian 11
 - Cent OS : compatibilité du déploiement sur la version 7
- Exploitation / Ansible : amélioration de la montée de version MongoDB suite à l'audit réalisé : prise en compte des clusters, définition des installations des packages pour se prémunir des montées de version non souhaitées

Griffons

- Montée de version de Siegfried (version 1.9.1 vers 1.9.6) et du référentiel PRONOM (version 109)

Sécurité

- Montée de version de COTS et de bibliothèques vers les dernières « bug fixe »
- Pour le module de collecte : amélioration par rapport à la version V6.RC. En V6 le module de collecte implémente la même séparation en couche de service externe / interne que les autres services vitam. Il est recommandé de porter une attention particulière à ce changement lors de la montée de version en V6 si vous avez installé le module de collecte en version V6.RC.

Modèle de données

La version 6.0 est rétrocompatible avec la version 6.RC. Suivent les ajouts faits pour la version 6.0 :

AJOUT : champ UnitUps, incluant les champs MetadataKey, MetadataValue et UnitUp dans la collection Project de la base de données Collect.

Workflow

La version 6 est rétrocompatible avec la version V6.RC sauf pour :

AJOUT : workflow d'entrée : contrôle de conformité à l'ontologie pour les métadonnées techniques étendues

API

Module de collecte : Mise à jour de la documentation technique des API (RAML)

- SUPPRESSION : API Dépréciée en V6.RC, rechercher des unités archivistiques d'un projet de versement (Path : /projects/{projectId}/units)
- MODIFICATION : si des attributs définis dans le projet ne sont pas renseignés lors de la création d'une transaction, alors par défaut la transaction créée doit contenir les valeurs du projet pour ces attributs (Path : /projects/{projectId}/transactions)

Recommandation d'usage des API collectes :

Pour des raisons de respect de logique métier ou pour des raisons de sécurité lors d'un usage en production, certains points d'API du module de collecte sont à réserver strictement à l'usage de l'application de gestion vitam-UI et ne doivent pas être autorisés aux applications versantes. La restriction d'habilitation des applications versantes se fait de la même manière que pour le reste de vitam. Par un acte d'exploitation, en associant au certificat de l'application versante un contexte applicatif qui référence un profil de sécurité définissant les permissions d'usage des différents points d'API. Ces points d'API sont identifiés dans la suite du document par la mention [<usage vitam-ui>](#).

/projects

/projects POST : [<usage vitam-ui>](#) création d'un projet de versement

/projects GET : [<usage vitam-ui>](#) recherche multicritère (titre et de service producteur) de projets sur un tenant, renvoie une liste /projects/<id>

/projects/<id> PUT : [<usage vitam-ui>](#) mise à jour d'un projet de versement identifié par son identifiant <id>

/projects/<id> DELETE : [<usage vitam-ui>](#) suppression d'un projet de versement identifié par son identifiant <id>

/projects/<id> GET : [<usage vitam-ui>](#) lecture d'un projet de versement identifié par son identifiant <id>

/projects/<id>/units

/projects/<id>/transaction

/projects/<id>/transaction POST : création d'une transaction dans projet de versement identifié par son identifiant <id>

`/projects/<id>/transaction` GET : [<usage vitam-ui>](#) Liste les transactions d'un projet de versement, renvoie une liste

`/transactions/<id>`

`/transactions/<id>` GET : retourne la transaction identifiée par son identifiant `<id>`

`/transactions/<id>` PUT : [<usage vitam-ui>](#) met à jour la transaction identifiée par son identifiant `<id>`

`/transactions/<id>` DELETE : [<usage vitam-ui>](#) supprime la transaction identifiée par son identifiant `<id>`

`/transactions/<id>/upload` POST : charge un lot d'archive dans la transaction à partir d'un ZIP contenant un CSV et des binaires.

`/transactions/<id>/close` POST : change le statut de la transaction, clôture la phase d'édition. La transaction est non modifiable.

`/transactions/<id>/reopen` POST : change le statut de la transaction, re-ouvre la phase d'édition. La transaction est modifiable.

`/transactions/<id>/send` POST : change le statut de la transaction, construit un SIP et le transfert à VITAM pour entrer.

`/transactions/<id>/abort` POST : change le statut de la transaction, force l'abandon d'une transaction qui sera purgée.

`/transactions/<id>/units`

`/transactions/<id>/units` POST : créé une unité d'archives dans une transaction identifiée par son identifiant `<id>`

`/transactions/<id>/units` GET : [<usage vitam-ui>](#) recherche multicritère (DSL) d'unité d'archives dans une transaction

`/units/<id>`

`/units/<id>` GET : [<usage vitam-ui>](#) retourne une unité d'archive identifiée par son identifiant `<id>`

`/units/<id>/<usage>/<version>`

`/units/<id>/<usage>/<version>` POST : créé ou met à jour un groupe d'objet technique pour l'usage version ciblé `<usage>`, `<version>`

`/units/<id>/<usage>/<version>/binary`

`/units/<id>/<usage>/<version>/binary` POST : charge un fichier (binaire) à associer à l'usage version ciblé `<usage>`, `<version>`

`/units/<id>/<usage>/<version>/binary` GET : [<usage vitam-ui>](#) retourne le fichier (binaire) associé à l'usage version donné `<usage>`, `<version>`

Montée de version / migration de données

Opération nécessaire pour toute montée de version 6.RC vers 6.0 : migration de données sur le module de collecte, procédure détaillée dans le document de montée de version

La migration des métadonnées étendues n'est pas prévue. En cas d'usage de métadonnées techniques hors SEDA sur les groupes d'objets, vous pouvez contacter le Programme Vitam.

Ajout de fonctionnalités

APP Collecte et préparation des versements :

- Affichage des métadonnées techniques
- Recherche sur des métadonnées externes via la liste de recherche « Autres critères » (acte d'exploitation pour paramétrer ces métadonnées)
- Mise à jour du projet de versement depuis le panneau latéral
- Création du projet de versement :
 - Modification des étapes de création d'un projet de versement manuel
 - Possibilité de créer un projet de versement en vue de paramétrer les versements d'un flux automatisé
 - Paramétrage du rattachement par clé/valeur dans un projet de versement

APP Dépôt et suivi des versements :

- Possibilité de cloisonner les accès aux versements par service(s) producteur(s)

APP Contrats de gestion (nouvelle APP) :

- Création de l'APP :
 - Accès au référentiel des contrats de gestion
 - Recherche et consultation d'un contrat de gestion
 - Création et, modification d'un contrat de gestion

APP Profils documentaires :

- Corrections d'anomalies

APP Recherche et consultation des archives :

- Affichage des métadonnées techniques
- Rechercher sur des métadonnées externes via la liste de recherche « Autres critères »
- Possibilité d'ajouter, modifier ou supprimer un blocage de règle (communicabilité, DUC, diffusion, réutilisation) sur une à plusieurs unités archivistiques

- Possibilité de bloquer ou débloquer un identifiant de règles sur une à plusieurs unités archivistiques (communicabilité, DUC, diffusion, réutilisation)

APP Contrats d'entrée :

- Ajout d'un profil utilisateur en vue de n'autoriser que la lecture du référentiel

APP Formats :

- Désactivation des boutons d'action pour un profil utilisateur n'ayant que des droits de consultation

Tests de non régression :

- Initialisation de tests de non régression automatisés

Ergonomie

- Ajout de pictogrammes matérialisant les arbres de positionnement, les plans de classement et les unités archivistiques standards dans la navigation arborescente

Documentation

- Ajout de la documentation APP Contrats de gestion

Exploitation

- Configuration par tenant d'une liste de métadonnées externes afin de pouvoir les sélectionner comme critères de recherche dans l'APP Recherche et consultation des archives