



VITAM - IHM Recette

Version 0.15.1

VITAM

avr. 26, 2017

1	Recette	1
1.1	Principes généraux	1
1.1.1	Avertissement	1
1.1.2	Accès	1
1.1.3	Titre des onglets	2
1.1.4	Sélection d'un tenant	2
1.1.5	Navigation	3
2	Administration des collections	5
2.1	Action de suppression	5
2.2	Purge de toutes les collections de la solution logicielle Vitam	5
2.3	Purge des référentiels	5
2.4	Purge des journaux	6
2.5	Purge des Unités Archivistiques et Groupes d'Objets	6
3	Tests SOAP-UI	7
3.1	Principes	7
3.2	Fichier de configuration	7
3.2.1	Structuration du fichier de configuration	8
3.2.2	Configuration d'un jeux de test	8
3.2.3	Exemple de configuration	10
3.2.4	Configuration de test pour api external avec multi-tenant	10
3.3	IHM	12
3.3.1	Lancement des tests	12
3.3.2	Affichage du rapport	13
3.3.3	Résultat au format JSON	13
4	Sécurisation des journaux	15
4.1	Lancer une opération de sécurisation	15
4.2	Journalisation des opérations de sécurisation	16
4.3	Rechercher des journaux sécurisés	16
5	Tests fonctionnels	19
5.1	Introduction	19
5.2	Page Tests Fonctionnels	19
5.3	Détail des tests	20

1.1 Principes généraux

1.1.1 Avertissement

L'IHM de recette est développée à des fins de tests uniquement.

Elle n'a aucunement vocation à être utilisée en production

Elle contient des interfaces utilisateurs permettant, par tenant, de :

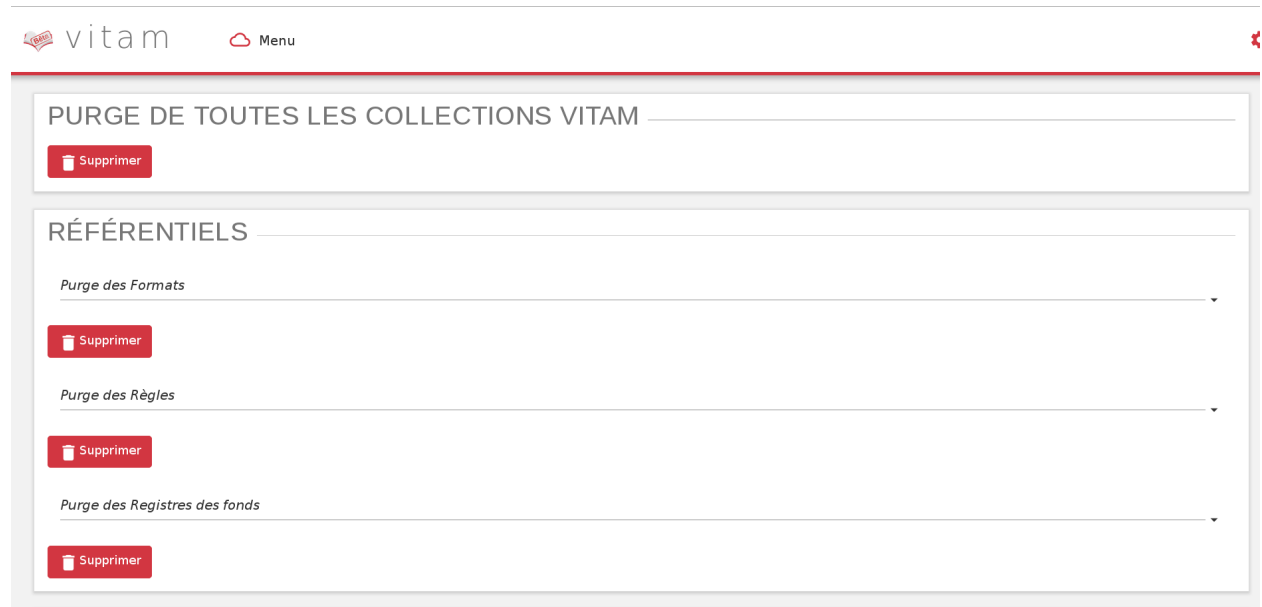
- Administrer les collections MongoDB (référentiels, journaux, objets...)
- Lancer des tests automatisés via SOAP UI
- Sécuriser manuellement les journaux des opérations
- Consulter les journaux de sécurisation

1.1.2 Accès

L'accès à l'IHM de recette s'effectue par un chemin différent de l'IHM démo. Par défaut, son adresse est :

adresse_de_votre_serveur/ihm-recette/#!/login

Par soucis de distinction visuelle avec l'interface de démo, la couleur dominante de cette IHM est le rouge.



1.1.3 Titre des onglets

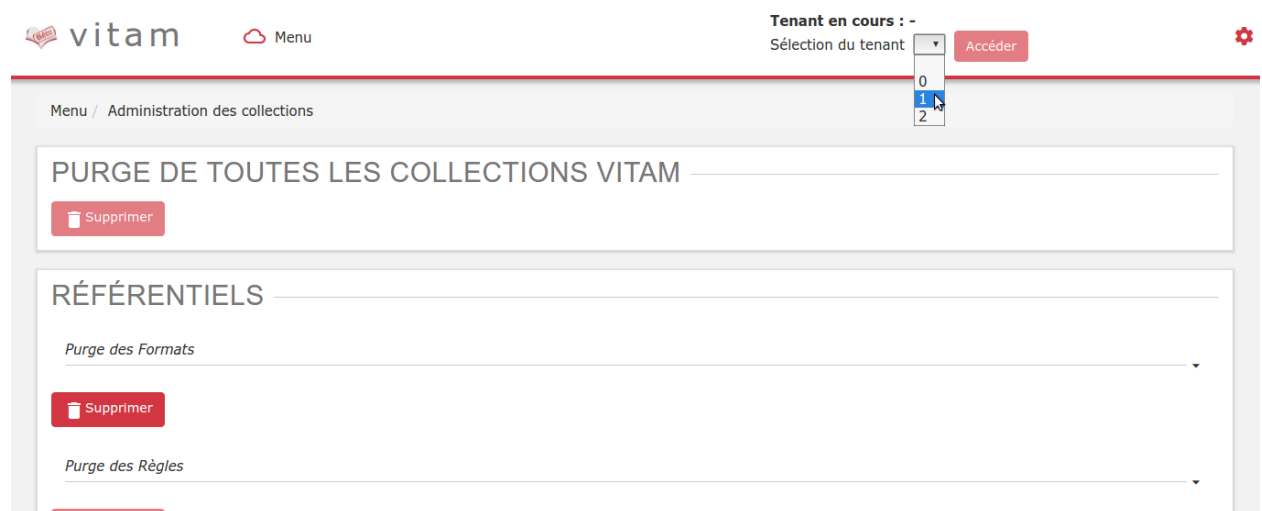
Sur IHM recette, le titre des pages est celui du dernier nœud du fil d'Ariane, précédé du mot "Recette -". Par exemple :

- Recette - Test SOAP-UI
- Recette - Administration des collections

1.1.4 Sélection d'un tenant

Lors de la connexion, l'utilisateur n'est positionné sur aucun tenant. De ce fait, ses actions d'administration sont restreintes car celles-ci sont pour la plupart liées à un tenant. Par défaut, un certain nombre de boutons sont donc grisés et inactifs.

Pour sélectionner un tenant, il suffit de choisir celui désiré dans le menu déroulant en haut à droite de l'écran et de valider sa sélection.



Une fois le tenant sélectionné, les boutons sont activés et l'intégralité de l'interface de recette est disponible.

Dans le reste de ce document, il est considéré que l'utilisateur s'est placé dans le tenant dans lequel il veut effectuer ses opérations. L'utilisateur peut changer de tenant à tout moment, en réitérant l'opération précédente.

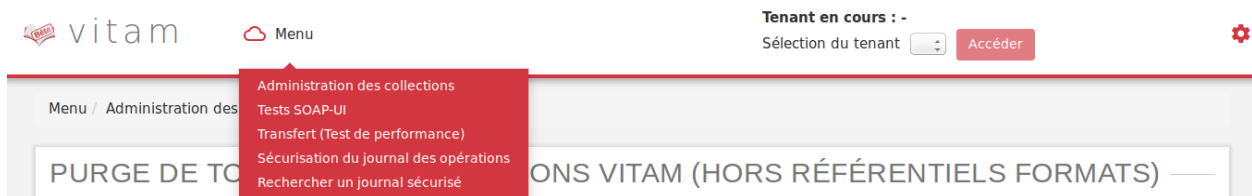
NB : le référentiel des formats est lié à la plateforme et non à un tenant. C'est pour cette raison que l'option de suppression du référentiel des formats est toujours disponible, même si aucun tenant n'est sélectionné.

1.1.5 Navigation

Par défaut, suite à sa connexion l'utilisateur accède à la page d'administration des collections.

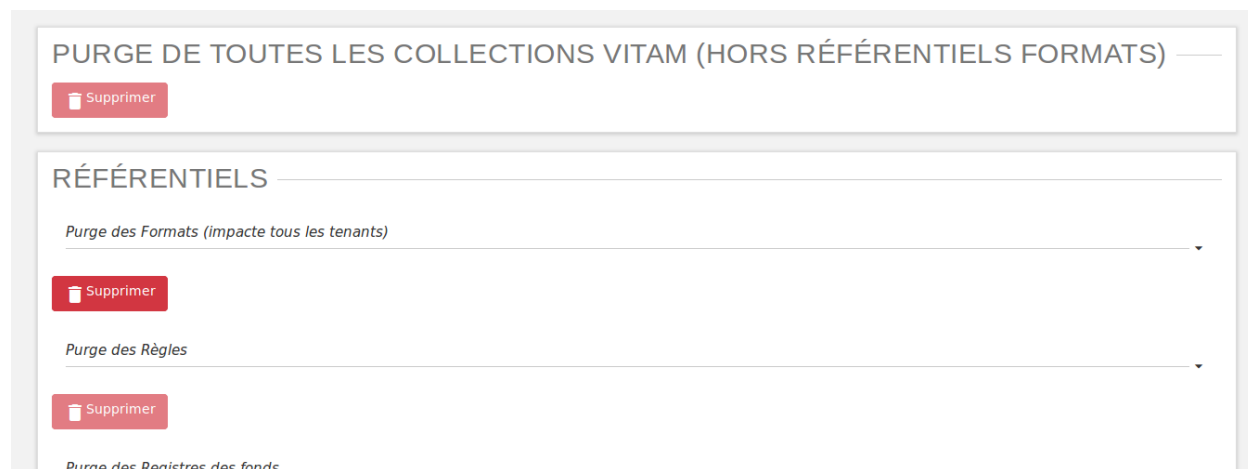
Le menu de navigation contient les entrées suivantes :

- Administration des collections
- Test SOAP-UI
- Transfert (test de performance)
- Sécurisation d'un journal des opérations
- Rechercher un journal sécurisé



Administration des collections

L'IHM de recette permet de supprimer certains ou tous les référentiels / journaux / objets dans le but de recetter VITAM ou de faire des tests variés pour éprouver la stabilité du système.



2.1 Action de suppression

Chaque bouton “supprimer” est situé sous le titre de son action. Lors du clic sur un des boutons “supprimer”, une fenêtre modale apparaît et demande de confirmer l’action de suppression. Deux choix sont proposés :

2.2 Purge de toutes les collections de la solution logicielle Vitam

La purge de toutes les collections correspond à la suppression de tous les référentiels, de tous les journaux ainsi que de tous les objets et unités archivistiques à l’exception du référentiel des formats. Suite à cette opération, chaque IHM correspondante est vide de contenu et plus aucune archive n’est présente dans VITAM.

2.3 Purge des référentiels

Il est possible de supprimer isolément un référentiel :

- Un clic sur “Annuler”, annule la demande de suppression, la fenêtre modale se ferme et rien ne se passe
- Un clic sur “Vider”, valide la demande de suppression, la fenêtre modale se ferme et la suppression est effectuée

Une fois la suppression effectuée, un message de confirmation s’affiche dans une fenêtre modale.

Référentiel des formats

Le référentiel des formats de la solution logicielle Vitam est supprimé **pour tous les tenants**. L’IHM du référentiel de formats est vide de contenu. Sans référentiel de formats, aucun SIP ne pourra être importé dans le SAE.

Référentiel des règles de gestion

Le référentiel des règles de gestion de la solution logicielle Vitam est supprimé. L’IHM des règles de gestion est vide de contenu. Sans référentiel des règles de gestion, aucun SIP ne pourra être importé dans le SAE.

Registre des fonds

Le contenu du registre des fonds de la solution logicielle Vitam est supprimé. L’IHM du registre des fonds est vide de contenu.

2.4 Purge des journaux

Journal du cycle de vie (unités archivistiques)

Tous les journaux du cycle de vie des unités archivistiques sont supprimés de la solution logicielle Vitam. L’IHM de chaque “Journal du cycle de vie” d’une unité archivistique est vide de contenu.

Journal du cycle de vie (objets)

Tous les journaux du cycle de vie des objets sont supprimés de la solution logicielle Vitam. L’IHM de chaque “Journal du cycle de vie” d’un objet est vide de contenu.

Journal des Opérations

Tous les journaux des opérations sont supprimés de la solution logicielle Vitam. Les IHM “Journal des opérations” et “Journal des opérations d’entrées” sont vides de contenu.

2.5 Purge des Unités Archivistiques et Groupes d’Objets

Purge des Unités Archivistiques

Toutes les unités archivistiques sont supprimées de la solution logicielle Vitam. L’IHM “Recherche d’archives” ne comprend plus d’objet.

Purge des groupes d’objets

Toutes les objets sont supprimés de la solution logicielle Vitam. L’IHM “Détail d’une unité archivistique” ne comprend plus d’objet.

Tests SOAP-UI

3.1 Principes

Cette interface a pour objectif de lancer des tests fonctionnels automatisés avec SOAP UI, construits via un fichier JSON.

L'équipe d'exploitation réalise le fichier Json avec les archivistes, puis le place dans la solution logicielle Vitam.

L'exécution des tests est ensuite lancée par les archivistes via l'IHM.

Cette partie présente à la fois l'IHM et les différentes configurations à réaliser par les archivistes.



3.2 Fichier de configuration

Les jeux de test à exécuter sont déclarés dans le fichier *data.json*.

Chaque SIP à utiliser pour valider un cas fonctionnel sera donc déclaré dans ce fichier, selon un méthode détaillée ci-après.

3.2.1 Structuration du fichier de configuration

Le fichier de configuration contient un champ “params” dont la valeur est un tableau. Ce tableau contient l’ensemble des jeux de tests à exécuter, chaque ligne représentant un test.

```
{ "params" :  
  [  
    {Configuration du test 1},  
    {Configuration du test 2},  
    {Configuration du test 3}  
  ]  
}
```

3.2.2 Configuration d’un jeux de test

Chaque jeux de test est composé des huit éléments suivants :

- sipName
- expected
- httpStatusCode
- testName
- category
- action [* name * expected

“sipName”

La valeur contenue dans ce champ doit être une chaîne de caractères. Il contient le nom du fichier à utiliser en entrée ainsi que sa localisation dans le système de fichier.

La localisation du fichier est une URI relative par rapport à la racine des tests SoapUI déterminée par votre exploitant.

On a donc par défaut l’arborescence suivante (le schéma suivant ne tient pas compte des autres dossiers et fichiers du repertoire SoapUI)

```
RACINE  
  |----- data.json  
  \----- test.zip
```

Pour un fichier *test.zip* placé dans le dossier *RACINE*, on entrera alors :

```
"sipName": "test.zip"
```

Si ce fichier est placé dans un dossier *data*, enfant de *RACINE*, c’est à dire placé comme ceci :

```
RACINE  
  |----- data.json  
  \----- data  
           \----- test.zip
```

on entrera alors :

```
"sipName": "test_arborescence/test.zip"
```

“expected”

La valeur contenue dans ce champ doit être une chaîne de caractères. Il contient le statut attendu à l’issue de l’opération d’entrée du SIP. La valeur de ce champ n’est utilisée que si les valeurs du tableau “Action” (détaillé plus bas) ne sont pas renseignées. Ceci permet de garder une compatibilité avec les anciens tests de non régression qui ne contenaient pas de tableau “Action”.

Les valeurs contenues dans ces champs sont, en revanche, ignorées si le tableau “Action” contient des valeurs.

Par exemple :

```
"expected": "OK",
```

“httpStatusCode”

La valeur contenue dans ce champs est un nombre entier. Il contient le code retour HTTP attendu.

Par exemple :

```
"httpStatusCode": 200,
```

“testName”

La valeur contenue dans ce champ doit être une chaîne de caractères. Il s’agit du nom du test tel qu’il sera reporté dans le rapport final.

Par exemple :

```
"Jeux de données possédant une arborescence complexe"
```

“category” : La valeur contenue dans ce champ doit être une chaîne de caractères.

Par exemple :

```
"category": "Tests sur les arborescences"
```

action : ce champ contient un tableau de n objets ayant pour but de tester des actions précises du workflow. Un objet est structuré de la façon suivante :

name : contient le nom de l’action à tester

expected : contient l’état attendu

voici l’exemple d’une configuration pour un jeu de test

NB : l’exemple de configuration d’un jeu de test ci-dessous a été indenté pour une meilleure compréhension. Par défaut, dans le fichier de configuration global, la configuration d’un test est placée sur une ligne seulement.

```
{
  "sipName": "SIP_OK/ZIP/OK_SIP_2_GO.zip",
  "expected": "OK",
  "httpStatusCode": 200,
  "testName": "Test des différentes étapes OK",
  "category": "Test global",
  "actions": [
    {
      "name": "UPLOAD_SIP",
      "expected": "OK"
    },
    {
```

```
    "name": "STP_SANITY_CHECK_SIP",
    "expected": "OK"
  },
}
```

3.2.3 Exemple de configuration

Ci-après un exemple de configuration contenant 4 jeux de tests

```
{
  "params": [
    {
      "sipName": "SIP_OK/ZIP/OK_SIP_2_GO.zip",
      "expected": "OK",
      "httpStatusCode": 200,
      "testName": "Test des différentes étapes OK",
      "category": "Test global",
      "actions": [
        {
          "name": "UPLOAD_SIP",
          "expected": "OK"
        },
        {
          "name": "STP_SANITY_CHECK_SIP",
          "expected": "OK"
        },
        {
          "name": "CHECK_SEDA",
          "expected": "OK"
        },
        {
          "name": "CHECK_MANIFEST_DATAOBJECT_VERSION",
          "expected": "OK"
        }
      ]
    },
    {
      "sipName": "SIP_KO/ZIP/KO_SIP_Mauvais_Format.pdf",
      "expected": "KO",
      "httpStatusCode": 200,
      "testName": "SIP au mauvais format",
      "category": "Test sur le Conteneur",
      "actions": [
        {
          "name": "UPLOAD_SIP",
          "expected": "KO"
        }
      ]
    }
  ]
}
```

3.2.4 Configuration de test pour api external avec multi-tenant

Chaque jeu de test est composé des éléments suivants :

- sipName / ruleName / formatName
- testName
- category
- tenant
- action [
 - name
 - endpoint
 - request
 - method
 - tenant
 - expected

“sipName / ruleName / formatName”

La valeur contenue dans ce champ doit être une chaîne de caractères. Il contient le nom du fichier à utiliser en entrée ainsi que sa localisation dans le système de fichier

La localisation du fichier est une URI relative par rapport à la racine des tests SoapUI déterminée par l’exploitant.

On a donc par défaut l’arborescence suivante : (le schéma suivant ne tient pas compte des autres dossiers et fichiers du repertoire SoapUI)

```
RACINE
|----- data.json
\----- SIP_KO
\----- SIP_OK
      \----- ZIP
              \----- OK_SIP.ZIP
      \----- TAR
\----- SIP_WARNING
\----- formats
\----- rules
      \----- regles_CSV.csv
```

Pour un SIP *OK_SIP.zip* et un fichier de règles de gestion *regles_CSV.csv* placés dans le dossier *RACINE*, on entrera alors :

```
"sipName": "SIP_OK\ZIP\OK_SIP.ZIP"
```

```
"ruleName": "rules\regles_CSV.csv"
```

“testName”

La valeur contenue dans ce champ doit être une chaîne de caractères. Il s’agit du nom du test tel qu’il sera reporté dans le rapport final. Par exemple :

```
"testName": "SRC1 : chercher des unités d’archives contenant des objets dans un_
↳ intervalle de dates extrêmes",
```

“category”

La valeur contenue dans ce champ doit être une chaîne de caractères. Elle doit toujours être qualifiée de : “Test API external”

“tenant”

La valeur contenue dans ce champ doit être une chaîne de caractères. Il s’agit du tenant qui a téléchargé le fichier SIP/règle/format. Par exemple :

```
"tenant" : "0"
```

“action”

ce champs contient un tableau de n objets ayant pour but de contrôler les réponses de l’API external. Un objet est structuré de la façon suivante :

- name** : contient le nom de l’action à tester
- endpoint** : contient l’endpoint de l’api external
- request** : contient la requête
- method** : contient la méthode
- tenant** : contient le tenant
- expected** : est un tableau qui contient le code retour HTTP attendu (httpStatusCode) et les champs attendus dans la réponse

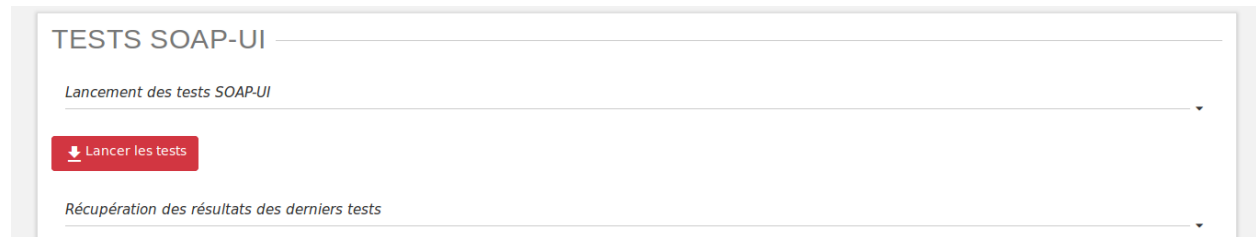
Par exemple :

```
"actions": [{
  "name": "SEARCH_RULES_TENANT_0",
  "endpoint" : "admin-external/v1/rules",
  "request" : "{ \"$query\": { \"$and\": [ { \"$eq\": { \"RuleValue\": \"Dossier individuel_
  ↪ d'agent civil\" } }, { \"$eq\": { \"RuleType\": \"AppraisalRule\" } } ] }, \"$filter\": {}, \"$
  ↪ $projection\": {} }",
  "method" : "POST",
  "tenant" : "0",
  "expected" : [ { "httpStatusCode": 200, "RuleId": [ "APP-00001" ] } ], {
  "name": "SEARCH_RULES_TENANT_1",
  "endpoint" : "admin-external/v1/rules",
  "request" : "{ \"$query\": { \"$and\": [ { \"$eq\": { \"RuleValue\": \"Dossier individuel_
  ↪ d'agent civil\" } }, { \"$eq\": { \"RuleType\": \"AppraisalRule\" } } ] }, \"$filter\": {}, \"$
  ↪ $projection\": {} }",
  "method" : "POST",
  "tenant" : "1",
  "expected" : [ { "httpStatusCode": 404 } ] }
```

3.3 IHM

3.3.1 Lancement des tests

Une fois le fichier *data.json*, ainsi que le SIP déposés sur le serveur Vitam par l’exploitant, les tests peuvent être lancés via l’IHM en cliquant sur le bouton “lancer les tests”



3.3.2 Affichage du rapport

Pour afficher le rapport, cliquer sur le bouton “Afficher résultats”.



Les deux tableaux de rapports apparaissent alors en bas de l'écran.

Le premier contient la synthèse des tests et est constitué des informations suivantes :

- Nombre de tests réalisés
- Nombre de tests en succès
- Nombre de tests en échec

RÉSULTATS DES DERNIERS TESTS

Résumé

Nb Tests	Succès	Echecs
5	3	2

Détails

ID Opération	Nom	Sip	Catégorie	Action	Attendu	Observé
aedqaaaaacaam7mxaagyoakzc7wx3byaaaaq	Test des différentes étapes OK	SIP_OK/ZIP/OK_SIP_2_GO.zip	Test global	UPLOAD_SIP	OK	
aedqaaaaacaam7mxaagyoakzc7wx3traaaaaq	SIP au mauvais format	SIP_KO/ZIP/KO_SIP_Mauvais_Format.pdf	Test sur le Conteneur	UPLOAD_SIP	KO	
aedqaaaaacaam7mxaagyoakzc7wx3byaaaaq	Test des différentes étapes OK	SIP_OK/ZIP/OK_SIP_2_GO.zip	Test global	STP_SANITY_CHECK_SIP	OK	OK
aedqaaaaacaam7mxaagyoakzc7wx3byaaaaq	Test des différentes étapes OK	SIP_OK/ZIP/OK_SIP_2_GO.zip	Test global	CHECK_SEDA	OK	OK
aedqaaaaacaam7mxaagyoakzc7wx3byaaaaq	Test des différentes étapes OK	SIP_OK/ZIP/OK_SIP_2_GO.zip	Test global	CHECK_MANIFEST_DATAOBJECT_VERSION	OK	OK

Le second contient le détail des tests et est constitué des colonnes suivantes :

- ID Opération
- Nom du test
- Nom du SIP
- Catégorie
- Action
- Résultat attendu
- Résultat constaté

Les lignes pour lesquelles le résultat attendu est égal au résultat constaté apparaissent sur fond vert. Les lignes pour lesquelles le résultat attendu est différent du résultat constaté apparaissent sur fond rouge

3.3.3 Résultat au format JSON

Il est également possible d'afficher le JSON de résultat en cliquant sur le lien nommé “Lien vers les résultats Json”. Le fichier apparaît en plein écran et il est possible de l'enregistrer.



Sécurisation des journaux

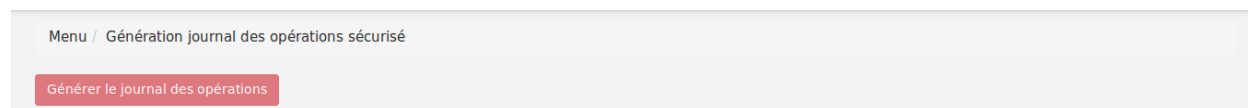
La sécurisation des journaux est une action visant à assurer la valeur probante de l'information prise en charge dans la solution logicielle Vitam.

4.1 Lancer une opération de sécurisation

L'interface de lancement est accessible par le menu : Menu > Sécurisation du journal des opérations

L'interface contient simplement un bouton "Générer le journal des opérations". Au clic sur ce bouton, le système va lancer l'opération de sécurisation des journaux. Elle prendra en compte tous les journaux du dernier créé au dernier non sécurisé.

Il est également possible d'afficher le JSON de résultat en cliquant sur le lien nommé "Lien vers les résultats Json". Le fichier apparaît en plein écran et il est possible de l'enregistrer.



Si aucun journal n'a encore été sécurisé, alors l'opération de sécurisation prendra en compte tous les journaux.

A la fin de l'opération, un message avertit du succès ou de l'échec de l'opération.

Un fichier .zip est créé et placé dans le l'offre de stockage de Vitam dans le répertoire suivant :

```
/browse/data/storage-offer-default/0/Logbook
```

Il contient les fichiers suivants :

- *operation.json* : liste des opérations sécurisées, la première étant l'opération "tracability"
- *merkleTree.json* : contient une sérialisation JSON de l'arbre de merkle
- *token.tsp* : horodatage de la combinaison de la racine de l'arbre de merkle, des empreintes des opérations de sécurisation antérieures (la dernière réalisée, celle du mois précédent et celle de l'année précédente)
- *computing_information.txt* : reprend les différentes empreintes qui ont permis de réaliser l'horodatage
- *additional_information.txt* : contient le nombre d'informations sécurisées, ainsi que les dates du premier et du dernier élément

4.2 Journalisation des opérations de sécurisation

La sécurisation des journaux des opérations donne lieu à la création d'un journal des opérations de type TRACEABILITY.

Ces journaux sont créés par tenant.

4.3 Rechercher des journaux sécurisés

L'interface de consultation des journaux sécurisés est accessible par le menu : Menu > Rechercher un journal sécurisé

L'interface est constituée de trois éléments :

- Un formulaire
- Un paginateur
- Une zone d'affichage des résultats

The screenshot shows a web interface for searching secured journals. At the top, there is a breadcrumb 'Menu / Recherche d'un journal sécurisé'. Below this is a search form titled 'RECHERCHE' with a dropdown arrow. The form contains three input fields: 'Identifiant de l'objet' (with an 'Effacer' button), 'Dates extrêmes' (with two date pickers), and 'Type de journal sécurisé' (with a dropdown menu). A 'Rechercher' button is located at the bottom right of the form. Below the search form is a paginator with 'Précédent', '0 / 0', and 'Suivant' buttons. At the bottom is a results table titled 'RÉSULTATS' with a dropdown arrow. The table has four columns: 'Type de journal sécurisé', 'Date de début', 'Date de fin', and 'Télécharger'.

Lancer une recherche

Par défaut, aucun résultat n'est affiché. Il faut lancer une recherche pour faire apparaître des résultats.

Pour lancer une recherche en prenant en compte cette date, cliquer sur le bouton "Rechercher" après l'avoir sélectionné.

Si l'utilisateur clique sur le bouton "Rechercher" sans sélectionner de date, alors tous les journaux disponibles s'affichent.

Utilisation du formulaire

Le formulaire est composé des champs suivants :

- Identifiant de l'objet : nom du fichier recherché
- Dates extrêmes : intervalle de dates permettant de rechercher sur les date de début d'un opération de sécurisation
- Type de journal sécurisé : liste déroulante permettant de cib



Zone de résultats

La zone de résultats est composée des colonnes suivantes :

- Type de journal sécurisé : affiche le type de journal sécurisé
- Date de début : indique la date de début de l'opération de sécurisation
- Date de fin : indique la date de fin de l'opération de sécurisation
- Télécharger : icône permettant de télécharger l'opération

Précédent /3 Suivant

RÉSULTATS (160)


Type de journal sécurisé	Date de début	Date de fin	Télécharger
OPERATION	03-04-2017 10:01	03-04-2017 11:01	
OPERATION	03-04-2017 09:01	03-04-2017 10:01	
OPERATION	03-04-2017 08:01	03-04-2017 09:01	
OPERATION	03-04-2017 07:01	03-04-2017 08:01	
OPERATION	03-04-2017 06:01	03-04-2017 07:01	
OPERATION	03-04-2017 05:01	03-04-2017 06:01	

Consultation du détail d'un journal

Chaque ligne représente un journal de type sécurisation. En cliquant sur cette ligne, l'IHM affiche le détail du journal dans une nouvelle fenêtre.

Téléchargement d'un journal

Chaque ligne représentant un journal comporte un symbole de téléchargement. En cliquant sur ce symbole, le journal est téléchargé sous forme de zip. Le nom de ce fichier correspond à la valeur du champ FileName du dernier event du journal de l'opération.

Type de journal sécurisé	Date de début	Date de fin	Télécharger
OPERATION	03-04-2017 10:01	03-04-2017 11:01	

Tests fonctionnels

5.1 Introduction

La partie “Tests Fonctionnels” contient les écrans de lancement et de consultation des résultats des TNR.

Elle est accessible depuis l’IHM de recette, par le menu Menu > Test Fonctionnels

Les tests ne sont pas segmentés par tenant. Ces derniers sont directement configurés dans les tests. Il n’est donc pas nécessaire de sélectionner un tenant pour accéder au contenu de cette section.

NB : La configuration des TNR ne s’effectue pas depuis ces écrans. La procédure de configuration est décrite dans la documentation dédiée à ce sujet.

5.2 Page Tests Fonctionnels

La page est divisée en deux parties :

- Boutons de gestion
- Résultats des derniers tests

Rapport	Détails
report_20170316_111400.json	Accès au détail
report_20170316_083754.json	Accès au détail

Boutons de gestion

- Bouton “Lancer les tests” : permet de rejouer les tests configurés. Ceci donnera lieu à la création d’un nouveau rapport.

- Bouton “Mise à jour référentiel” : permet de récupérer les derniers fichiers de configuration des tests depuis git (gestionnaire de sources). Ainsi, si un utilisateur a ajouté des tests et que ceux-ci ont été intégrés à git, le fait de cliquer sur ce bouton permettra de les prendre en compte au prochain clic sur le bouton “Lancer les Tests”.

Résultat des derniers tests

Les résultats de tests sont affichés dans un tableau à deux colonnes :

- Rapport
- Détail

Chaque ligne représente le rapport issu d’une campagne de tests. La colonne rapport indique le nom du rapport. Celui-ci est constitué de la façon suivante : report_AAAAMMJJ_HHmms.json. Ainsi le rapport correspondant à la dernière campagne de tests se trouve au-dessus de la liste.

La colonne détail affiche simplement la mention “accès au détail”.

Au clic sur une ligne, la page du détail du rapport concerné s’affichera dans un nouvel onglet.

5.3 Détail des tests

L’écran de détail d’une campagne de tests est divisé en deux parties :

- Partie Résumé
- Partie Détails

Feature	ID Opération	Description	Errors
✖ Calcul des règles de gestion		Recherche une archive unit avec les règles héritées en cas de la prévention d'héritage (PreventInheritance)	java.nio.file.NoSuchFileException: /vitam/data/ihm-recette/test-data/system/data/SIP_OK /ZIP/1066_CA6_corrige.zip at sun.nio.fs.UnixException.translateToIOException(UnixException.java:86) at sun.nio.fs.UnixException.rethrowAsIOException(UnixException.java:102) at sun.nio.fs.UnixException.rethrowAsIOException(UnixException.java:107) at sun.nio.fs.UnixFileSystemProvider.newByteChannel(UnixFileSystemProvider.java:214) at java.nio.file.Files.newByteChannel(Files.java:361) at java.nio.file.Files.newByteChannel(Files.java:407) at java.nio.file.spi.FileSystemProvider.newInputStream(FileSystemProvider.java:384) at java.nio.file.Files.newInputStream(Files.java:152) at fr.gouv.vitam.functionaltest.cucumber.step.IngestStep.upload_this_sip(IngestStep.java:90) at ✖.Quand je

Partie Résumé



La partie résumé comporte les trois indications suivantes :

- NB Tests : nombre de tests inclus dans la campagne
- Succès : nombre de tests en succès
- Echecs : nombre de tests en échec

Partie Détails

Chaque ligne du tableau représente le résultat d’un test. Celle-ci est sur fond vert lorsque le test est en succès, sur fond rouge lorsqu’il est en échec.

Ci-après l’exemple de lignes correspondant à un test en succès.

	uploader des fichier SIP	aedqaaaaacgbcaacaajvsak22er2kzyaaaaq	SIP au mauvais format
	uploader des fichier SIP	aedqaaaaacgbcaacaajvsak22er3dnyaaaaq	Test virus

Le tableau est constitué de cinq colonnes :

- Icône : Correspond au statut du test. Il s'agit d'une coche verte pour un succès, d'une croix rouge pour un échec.
- feature : Correspond à la fonctionnalité testée. Par défaut, un fichier de configuration correspond à une fonctionnalité. On a par exemple un fichier de configuration pour réaliser tous les tests sur l'INGEST. Dans ce cas, le nom de la fonctionnalité sera indiqué dans tous les cas de test correspondant dans le tableau de restitution.
- ID opération : Identifiant de l'opération correspondant au test. On peut ainsi s'en servir pour trouver plus de détails sur le test dans le journal des opérations.
- Description : Il s'agit d'une description du cas de test effectué. Celle-ci est indiquée dans le fichier de configuration pour chacun des tests.
- Errors : Erreur technique liée à l'échec du test. Cette colonne est vide pour les tests en succès.