VITAM - Modèle de données

Version 0.20.0

VITAM

juil. 21, 2017

Table des matières

1 Introduction **1**
- 1.1 Avertissement . . . 1
- 1.2 Objectif du document . . . 1
- 1.3 Création des index . . . 1
- 1.4 Généralités . . . 1
  - 1.4.1 Nommage des champs . . . 1

2 Base Logbook **3**
- 2.1 Collections contenues dans la base . . . 3
- 2.2 Collection LogbookOperation . . . 3
  - 2.2.1 Utilisation de la collection LogbookOperation . . . 3
  - 2.2.2 Exemple de JSON stocké dans la collection . . . 3
  - 2.2.3 Détail des champs du JSON stocké dans la collection . . . 5
  - 2.2.4 Détail des champs du JSON stocké en base spécifiques à une opération de sécurisation . . . 7
- 2.3 Collection LogbookLifeCycleUnit . . . 8
  - 2.3.1 Utilisation de la collection LogbookLifeCycleUnit . . . 8
  - 2.3.2 Exemple de JSON stocké en base . . . 8
  - 2.3.3 Détail des champs du JSON stocké en base . . . 9
  - 2.3.4 Détail des champs du JSON stocké en base spécifiques à une mise à jour . . . 11
- 2.4 Collection LogbookLifeCycleObjectGroup . . . 11
  - 2.4.1 Utilisation de la collection LogbookLifeCycleObjectGroup . . . 11
  - 2.4.2 Exemple de JSON stocké en base . . . 11
  - 2.4.3 Détail des champs du JSON stocké en base . . . 12

3 Base MetaData **15**
- 3.1 Collections contenues dans la base . . . 15
- 3.2 Collection Unit . . . 15
  - 3.2.1 Utilisation de la collection Unit . . . 15
  - 3.2.2 Exemple de JSON . . . 15
  - 3.2.3 Exemple de XML en entrée . . . 16
  - 3.2.4 Détail du JSON . . . 17
- 3.3 Collection ObjectGroup . . . 18
  - 3.3.1 Utilisation de la collection ObjectGroup . . . 18
  - 3.3.2 Exemple de Json stocké en base . . . 18
  - 3.3.3 Exemple de XML . . . 20
  - 3.3.4 Détail des champs du JSON . . . 21

4 Base MasterData **23**

4.1 Collections contenues dans la base . . . 23
4.2 Collection FileFormat . . . 23
4.2.1 Utilisation de la collection FileFormat . . . 23
4.2.2 Exemple de JSON stocké en base . . . 23
4.2.3 Exemple de la description d'un format dans le XML d'entrée . . . 24
4.2.4 Détail des champs du JSON stocké en base . . . 24
4.3 Collection FileRules . . . 25
4.3.1 Utilisation de la collection FileRules . . . 25
4.3.2 Exemple de JSON stocké en base . . . 26
4.3.3 Structure du fichier d'import . . . 26
4.3.4 Détail des champs . . . 26
4.4 Collection IngestContract . . . 27
4.4.1 Utilisation de la collection . . . 27
4.4.2 Exemple de JSON stocké en base . . . 27
4.4.3 Exemple d'un fichier d'import de contrat . . . 28
4.4.4 Détail des champs . . . 28
4.5 Collection AccessContract . . . 29
4.5.1 Utilisation de la collection . . . 29
4.5.2 Exemple de JSON stocké en base . . . 29
4.5.3 Exemple d'un fichier d'import de contrat d'accès . . . 29
4.5.4 Détail des champs . . . 30
4.6 Collection Profile . . . 30
4.6.1 Utilisation de la collection . . . 30
4.6.2 Exemple de JSON stocké en base . . . 30
4.6.3 Exemple d'un fichier d'import de profils . . . 31
4.6.4 Détail des champs . . . 31
4.7 Collection Context . . . 32
4.7.1 Utilisation de la collection . . . 32
4.7.2 Exemple de JSON stocké en base . . . 32
4.7.3 Détail des champs . . . 33
4.8 Collection AccessionRegisterSummary . . . 33
4.8.1 Utilisation de la collection . . . 33
4.8.2 Exemple de JSON stocké en base . . . 33
4.8.3 Exemple de la description dans le XML d'entrée . . . 34
4.8.4 Détail des champs . . . 34
4.9 Collection AccessionRegisterDetail . . . 35
4.9.1 Utilisation de la collection . . . 35
4.9.2 Exemple de JSON stocké en base . . . 35
4.9.3 Exemple de la description dans le XML d'entrée . . . 36
4.9.4 Détail des champs . . . 36
4.10 Collection VitamSequence . . . 38
4.10.1 Utilisation de collection . . . 38
4.10.2 Exemple de JSON stocké en base . . . 38
4.10.3 Détail des champs . . . 38

5 Annexes **39**
5.1 Valeurs possibles pour le champ evType du LogBook Operation . . . 39
5.2 Valeurs possibles pour le champ evType du LogBook LifeCycle . . . 39
5.3 Valeurs possibles pour le champ evTypeProc . . . 39
5.4 Catégories de règles possibles . . . 39
5.5 Valeurs possibles pour le champ Status de la collection AccessionRegisterDetail . . . 40
5.6 Valeurs possibles pour le champ Name de la collection VitamSecquence . . . 40

CHAPITRE 1

Introduction

1.1 Avertissement

Ce document fait état du travail en cours. Il est susceptible de changer de manière conséquente.

1.2 Objectif du document

Ce document a pour objectif de présenter la structure générale des collections utilisées dans la solution logicielle Vitam. Il est destiné principalement aux développeurs, afin de leur présenter l'organisation des données dans la solution logicielle Vitam, ainsi qu'à tous les autres acteurs du programme pour leur permettre de connaître ce qui existe en l'état actuel.

Il explicite chaque champ, précise la relation avec les sources (manifest conforme au standard SEDA v.2.0, référentiels Pronom, etc...) et la structuration JSON stockée dans la base de données MongoDB.

Pour chacun des champs, cette documentation apporte :

- Une liste des valeurs licites
- La sémantique ou syntaxe du champ
- La codification en JSON

Il décrit aussi parfois une utilisation particulière faite à une itération donnée. Cette indication diffère de la cible finale, auquel cas le numéro de l'itération de cet usage est mentionné.

1.3 Création des index

Les différents index sont créés par ansible. Les fichiers à renseigner pour rajouter un nouvel index sont stockés dans le répertoire deployment/ansible-vitam/roles/mongo_configure/templates/init-{nom-base}-database.js.j2

1.4 Généralités

1.4.1 Nommage des champs

Les champs des fichiers JSON présents dans les collections peuvent être nommés de deux manières :

- "champ" : un champ sans préfixe est modifiable via les API.
- "_champ" : un champ ayant pour préfixe "_" n'est pas modifiable via les API.

CHAPITRE 2

Base Logbook

2.1 Collections contenues dans la base

La base Logbook contient les collections relatives aux journaux d'opérations et de cycles de vie des unités archivistiques et des objets numériques de la solution logicielle Vitam.

2.2 Collection LogbookOperation

2.2.1 Utilisation de la collection LogbookOperation

La collection LogbookOperation comporte toutes les informations de traitement liées aux opérations effectuées dans la solution logicielle Vitam, chaque opération faisant l'objet d'un enregistrement distinct.

Ces opérations sont :

- Entrée (implémentée dans la release en cours)
- Mise à jour (implémentée dans la release en cours)
- Données de référence (implémentée dans la release en cours)
- Audit (non implémentée dans la release en cours)
- Elimination (non implémentée dans la release en cours)
- Préservation (non implémentée dans la release en cours)
- Vérification (implémentée dans la release en cours)
- Sécurisation (implémentée dans la release en cours)

Les valeurs correspondant à ces opérations dans les journaux sont détaillées dans l'annexe 5.3.

2.2.2 Exemple de JSON stocké dans la collection

Extrait d'un JSON correspondant à une opération d'entrée terminée avec succès.


```
{
   "_id": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
   "evId": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
   "evType": "PROCESS_SIP_UNITARY",
   "evDateTime": "2017-06-29T09:23:21.142",
   "evDetData": "{\"evDetDataType\":\"MASTER\",\"EvDetailReq\":\"Jeu de test avec␣
↪arborescence complexe\",\"EvDateTimeReq\":\"2016-11-22T13:50:57\",\
↪"ArchivalAgreement\":\"ArchivalAgreement0\",\"AgIdTrans\":\"Identifier5\"}",
```

```
  "evIdProc": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
  "evTypeProc": "INGEST",
  "outcome": "STARTED",
  "outDetail": "PROCESS_SIP_UNITARY.STARTED",
  "outMessg": "Début du processus d'entrée du SIP :␣
↪aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
  "agId": "{\"Name\":\"vitam-iaas-app-01\",\"Role\":\"ingest-external\",\"ServerId\":
↪1048375580,\"SiteId\":1,\"GlobalPlatformId\":243069212}",
  "agIdApp": null,
  "agIdAppSession": null,
  "evIdReq": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
  "agIdSubm": null,
  "agIdOrig": null,
  "obId": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
  "obIdReq": null,
  "obIdIn": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
  "events": [
      {
          "evId": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
          "evType": "PROCESS_SIP_UNITARY",
          "evDateTime": "2017-06-29T09:23:21.142",
          "evDetData": null,
          "evIdProc": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
          "evTypeProc": "INGEST",
          "outcome": "STARTED",
          "outDetail": "PROCESS_SIP_UNITARY.STARTED",
          "outMessg": "Début du processus des contrôles préalables à l'entrée",
          "agId": "{\"Name\":\"vitam-iaas-app-01\",\"Role\":\"ingest-external\",\
↪"ServerId\":1048375580,\"SiteId\":1,\"GlobalPlatformId\":243069212}",
          "evIdReq": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
          "obId": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
          "obIdReq": null,
          "obIdIn": null
      },
      {
          "evId": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
          "evType": "SANITY_CHECK_SIP",
          "evDateTime": "2017-06-29T09:23:21.206",
          "evDetData": null,
          "evIdProc": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
          "evTypeProc": "INGEST",
          "outcome": "STARTED",
          "outDetail": "SANITY_CHECK_SIP.STARTED",
          "outMessg": "Début du contrôle sanitaire",
          "agId": "{\"Name\":\"vitam-iaas-app-01\",\"Role\":\"ingest-external\",\
↪"ServerId\":1048375580,\"SiteId\":1,\"GlobalPlatformId\":243069212}",
          "evIdReq": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
          "obId": "aedqaaaaachhz4i4abx24ak46muxxkaaaaaq",
          "obIdReq": null,
          "obIdIn": null
      },
      {
          [...]
      }
  ],
  "_tenant": 0
}
```

2.2.3 Détail des champs du JSON stocké dans la collection

Chaque entrée de cette collection est composée d'une structure auto-imbriquée : la structure possède une première instanciation "incluante", et contient un tableau de N structures identiques, dont seules les valeurs contenues dans les champs changent.

La structure est décrite ci-dessous. Pour certains champs, on indiquera s'il s'agit de la structure incluante ou d'une structure incluse dans celle-ci.

"_id" (identifier) : Identifiant unique donné par le système lors de l'initialisation de l'opération Il s'agit d'une chaîne de 36 caractères correspondant à un GUID. Cet identifiant constitue la clé primaire de l'opération dans la collection.

Ce champ existe uniquement pour la structure incluante.

"evId" (event Identifier) : identifiant de l'événement Il s'agit d'une chaîne de 36 caractères. Il identifie l'opération de manière unique dans la collection. Cet identifiant doit être l'identifiant d'un événement dans le cadre de l'opération (evIdProc) et doit donc être différent par paire (début/fin).

Ce champ existe pour les structures incluantes et incluses

"evType" (event Type) : nom de l'événement Issu de la définition du workflow en JSON (fichier default-workflow.json). La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code est stocké dans ce champ, la traduction se faisant via un fichier properties (vitam-logbook-message-fr.properties).

Ce champ existe pour les structures incluantes et incluses

"evDateTime" (event DateTime) : date de l'événement Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+"T"+hh :mm :ss :[3digits de millisecondes] Positionnée par le client LogBook. `Exemple : "2016-08-17T08:26:04.227"`

Ce champ existe pour les structures incluantes et incluses

"evDetData" (event Detail Data) : détails de l'événement. Donne plus de détail sur l'événement ou son résultat. Par exemple, pour l'étape ATR_NOTIFICATION, ce champ détaille le nom de l'ArchiveTransferReply, son empreinte et l'algorithme utilisé pour calculer l'empreinte. Sur la structure incluante du journal d'opérations d'entrée, il contient un JSON composé des champs suivants :

- evDetDataType : structure impactée. Chaîne de caractères. Doit correspondre à une valeur de l'énumération LogbookEvDetDataType
- EvDetailReq : précisions sur la demande de transfert. Chaîne de caractères. Reprend le champ "Comment" du message ArchiveTransfer.
- EvDateTimeReq : date de la demande de transfert inscrit dans le champs evDetData. Date au format ISO8601 AAAA-MM-JJ+"T"+hh :mm :ss :[3digits de millisecondes].
- ArchivalAgreement : contrat d'entrée. chaîne de caractères. Reprend le nom du contrat utilisé pour réaliser l'entrée, indiqué dans le champ ArchivalAgreement du message ArchiveTransfer
- AgIfTrans : Service versant. chaîne de caractères. Contient le nom de l'entité ayant réalisé le transfert du SIP. Reprend le contenu du champ TransferringAgency du message ArchiveTransfer
- ServiceLevel : niveau de service. Chaîne de caractères. Reprend le champ ServiceLevel du message ArchiveTransfer

Ce champ existe pour les structures incluantes et incluses

"evIdProc" (event Identifier Process) : identifiant du processus. Il s'agit d'une chaîne de 36 caractères. Toutes les mêmes entrées du journal des opérations partagent la même valeur, qui est celle du champ "_id"

Ce champ existe pour les structures incluantes et incluses

"evTypeProc" (event Type Process) : type de processus. Il s'agit d'une chaîne de caractères. Nom du processus qui effectue l'action, parmi une liste de processus possibles fixée. Cette liste est disponible en annexe 5.3.

Ce champ existe pour les structures incluantes et incluses

"outcome" : Statut de l'événement. Il s'agit d'une chaîne de caractères devant correspondre à une valeur de la liste suivante :

- STARTED (début de l'événement)
- OK (Succès de l'événement)
- KO (Echec de l'événement)
- WARNING (Succès de l'événement comportant des alertes)
- FATAL (Erreur technique)

Ce champ existe pour les structures incluantes et incluses

"outDetail" (outcome Detail) : code correspondant au résultat de l'événement. Il s'agit d'une chaîne de caractères. Il contient le code correspondant au résultat de l'événement, incluant le statut. La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code doit être stocké dans ce champ, la traduction doit se faire via un fichier properties (vitam-logbook-message-fr.properties)

Ce champ existe pour les structures incluantes et incluses

"outMessg" (outcomeDetailMessage) : détail de l'événement. Il s'agit d'une chaîne de caractères. C'est un message intelligible destiné à être lu par un être humain en tant que détail de l'événement. Traduction du code présent dans outDetail issue du fichier vitam-logbook-message-fr.properties.

Ce champ existe pour les structures incluantes et incluses

"agId" (agent Identifier) : identifiant de l'agent interne réalisant l'action. Il s'agit de plusieurs chaînes de caractères indiquant le nom, le rôle et le PID de l'agent. Ce champ est calculé par le journal à partir de ServerIdentifier. `Exemple : {\"name\":\"ingest-internal_1\",\"role\":\"ingest-internal\",\"pid\":425367}`

Ce champ existe pour les structures incluantes et incluses

"agIdApp" (agent Identifier Application) : identifiant de l'application externe qui appelle la solution logicielle Vitam pour effect

Actuellement, la valeur est toujours 'null' mais sera renseignée une fois le mécanisme d'authentification mis en place. Ce champ existe uniquement pour la structure incluante.

"agIdAppSession" (agent Identifier Application Session) : identifiant donnée par l'application utilisatrice externe qui appelle la

L'application externe est responsable de la gestion de cet identifiant. Il correspond à un identifiant pour une session donnée côté application externe.

Actuellement, la valeur est toujours 'null' mais sera renseignée une fois le mécanisme d'authentification mis en place. Ce champ existe pour les structures incluantes et incluses

"evIdReq" (event Identifier Request) : identifiant de la requête déclenchant l'opération. Il s'agit d'une chaîne de 36 caractères. Une requestId est créée pour chaque nouvelle requête http venant de l'extérieur. Dans le cas du processus d'entrée, il devrait s'agir du numéro de l'opération (EvIdProc).

Ce champ existe pour les structures incluantes et incluses. Il s'agit du X-Application-Id.

"agIdSubm" (agent Identifier Submission) : identifiant du service versant. Il s'agit d'une chaîne de caractères. Reprend le contenu du champ SubmissionAgencyIdentifier du message ArchiveTransfer. Non rempli pour l'instant.

Ce champ existe uniquement pour la structure incluante.

"agIdOrig" (agent Identifier Originating) : identifiant du service producteur. Il s'agit d'une chaîne de caractères. Reprend le contenu du champ OriginatingAgencyIdentifier du message ArchiveTransfer.

Ce champ existe uniquement pour la structure incluante.

"obId" (object Identifier) : identifiant Vitam du lot d'objets auquel s'applique l'opération (lot correspondant à une liste).

Il s'agit d'une chaîne de 36 caractères. Dans le cas d'une opération d'entrée, il s'agit du GUID de l'entrée (evIdProc). Dans le cas d'une opération 'Audit', il s'agit par exemple du nom d'un lot d'archives prédéfini

Ce champ existe pour les structures incluantes et incluses

"obIdReq" (object Identifier Request) : Identifiant Vitam de la requête caractérisant un lot d'objets auquel s'applique l'opérati

Ne concerne que les lots d'objets dynamiques, c'est-à-dire obtenus par la présente requête. Ne concerne pas les lots ayant un identifiant défini.

Actuellement, la valeur est toujours 'null'. Ce champ existe pour les structures incluantes et incluses

"obIdIn" (ObjectIdentifierIncome) : Identifiant externe du lot d'objets auquel s'applique l'opération. Chaîne de caractère intelligible pour un humain qui permet de comprendre à quel SIP ou quel lot d'archives se raporte l'événement. Reprend le contenu du champ MessageIdentifier du message ArchiveTransfer.

Ce champ existe pour les structures incluantes et incluses Non utilisé à ce jour

"events" : tableau de structure. Pour la structure incluante, le tableau contient N structures incluses dans l'ordre des événements (date)

Ce champ existe uniquement pour la structure incluante.

"_tenant" : identifiant du tenant. Il s'agit d'un entier. *Ce champ existe uniquement pour la structure incluante.*

2.2.4 Détail des champs du JSON stocké en base spécifiques à une opération de sécurisation

Exemple de données stockées :


```
"evDetData":
{
\"LogType\":\"OPERATION\",
\"StartDate\":\"2017-06-29T09:22:23.227\",
\"EndDate\":\"2017-06-29T09:39:08.690\",
\"Hash\":\
 ↪"HYnFf07gFkar3lO+U2FQ9qkhi9eUMFN5hcH7oU7vrAAL3FAlMm8aJP7+VxkVWhLzmmFolwUEcq6fbS7Km2is5g==\
 ↪",
\"TimeStampToken\":\
 ↪"MIIEljAVAgEAMBAMDk9wZXJhdGlvbiBPa2F5MIIEewYJKoZIhvcNAQcCoIIEbDCCBGgCAQMxDzANBglghkgBZQMEAgMFADCBg
 ↪rIVKZ74J09qdSDeHw24HHsjw0tAnHjD6ZfUJHjDp8yQSdB6Lf2a6ORPF5JCgsh86CctQ9h93mwIBARgPMjAxNzA2MjkwOTM5MD
 ↪IdBglcs69fsH05yuXOEYuwPhN1yQijSGEwZAYLKoZIhvcNAQkQAi8xVTBTMFEwTzALBglghkgBZQMEAgMEQGlkJQTJOiVJrGpF
 ↪/wOcpCmpqIET8w2yUcP1yqQJXYc87YeY1/OWhZiWFqbWXVV9HS4wDQYJKoZIhvcNAQENBQAEggIAV/
 ↪rdnxIAyhvoGDprIahKAK3TPcriTggh1+gtDjEiD7kGB0KtXwAmPn2gb/2YtOmvIU7/
 ↪a5KBFlfBR+foIRrc6z52cEdalhSpyHpYgpFuF7SjMFO6Mfso1dwjI9KpZTv6OI6Kplbg6zwK939GpDbPgKaMrXw0EDafk184RQ
 ↪LbBEOUswGgnfnYGOlo1XbQaI2sM2+YiXHGD/qnl/
 ↪uAteBayFeaHKXel+gkp8D1ykBFOrE46n6fCI5i0OhKHcPAxvxTg8p03M38PrZIwnqSUI1rxfJhk9Hu0JVcQi1EYLBMmyL4IbhXI
 ↪BGTMZmuEksrA4vJr1WEFMUocEFQnL9pOJ+iI8U0SusJEDYvjde+yvfnxC8ZOGXOsaP9aUsuITOMT/
 ↪wFdrH4RFe8q8Wjxzu5p4lSvJI9P+soSfBbLyzGUjmF2lAi/Hdyzjunhmrr/
 ↪kxHK8P9Bo2CSz77xgN566k2r44ER/
 ↪lyHFvHme5ITq25CRhJf39kfbPh1Jjku3ulwiquykhnjXX7YGx5bDRNv+z29l4tq+AkZqq8O+0XY5fLGgauptsKhpj+CsfYs0uNa
 ↪rKg=\",
\"NumberOfElement\":379,
\"FileName\":\"0_LogbookOperation_20170629_093907.zip\",
\"Size\":3975227,
\"DigestAlgorithm\":\"SHA512\"}"
```


Dans le cas de l'événement final d'une opération de sécurisation du LogbookOperation, le champ **"evDetData"** est composé des champs suivants :

"LogType" : type de logbook sécurisé. Collection faisant l'objet de l'opération de sécurisation (LogbookOperation) `Exemple : "operation"`

"StartDate" : Date de début de la période de couverture de l'opération de sécurisation. Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+"T"+hh :mm :ss :[3digits de millisecondes] (correspond à la date de la dernière opération sécurisée par la précédente sécurisation) `Exemple : "2016-08-17T08:26:04.227"`

"EndDate" : date de fin de la période de couverture de l'opération de sécurisation. Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+"T"+hh :mm :ss :[3digits de millisecondes] (correspond à la date de la dernière opération sécurisée par la précédente sécurisation) `Exemple : "2016-08-17T08:26:04.227"`

"PreviousLogbookTraceabilityDate" : date de la précédente opération de sécurisation. Il s'agit de la date de début de la précédente opération de sécurisation du même type au format ISO8601 AAAA-MM-

JJ+"T"+hh :mm :ss :[3digits de millisecondes] (correspond à la date de début de la sécurisation précédente) `Exemple : "2016-08-17T08:26:04.227"`

"MinusOneMonthLogbookTraceabilityDate" : date de l'opération de sécurisation passée d'un mois. Il s'agit de la date de début de la précédente opération de sécurisation du même type réalisée un mois avant au format ISO8601 AAAA-MM-JJ+"T"+hh :mm :ss :[3digits de millisecondes] `Exemple : "2016-08-17T08:26:04.227"`

"MinusOneYeargbookTraceabilityDate" : date de l'opération de sécurisation passée d'un an. Il s'agit de la date de début de la précédente opération de sécurisation du même type réalisée un an avant au format ISO8601 AAAA-MM-JJ+"T"+hh :mm :ss :[3digits de millisecondes] `Exemple : "2016-08-17T08:26:04.227"`

"Hash" : Empreinte racine. Il s'agit d'une chaîne de caractères. Empreinte de la racine de l'arbre de Merkle.

"TimeStampToken" : Tampon d'horodatage. Il s'agit d'une chaîne de caractères. Tampon d'horodatage sûr du journal sécurisé.

"NumberOfElement" : Nombre d'éléments. Il s'agit d'un entier. Nombre d'opérations sécurisées.

"Size" : Taille du fichier. Il s'agit d'un entier. Taille du fichier sécurisé (en bytes).

"FileName" : Identifiant du fichier. Il s'agit d'une chaîne de caractères. Nom du fichier sécurisé dans le stockage au format {tenant}_LogbookOperation_{AAAAMMJJ_HHMMSS}.zip. `Exemple : "0_LogbookOperation_20170127_141136.zip"`

"DigestAlgorithm" : algorithme de hachage. Il s'agit d'une chaîne de caractères. Il s'agit du nom de l'algorithme de hachage utilisé pour réaliser le tampon d'horodatage.

2.3 Collection LogbookLifeCycleUnit

2.3.1 Utilisation de la collection LogbookLifeCycleUnit

Le journal du cycle de vie d'une unité archivistique (ArchiveUnit) trace tous les événements qui impactent celle-ci dès sa prise en charge dans le système. Il doit être conservé aussi longtemps que l'ArchiveUnit est gérée par le système.

- dès la réception d'une ArchiveUnit, on trace les opérations effectuées sur elles
- les journaux du cycle de vie sont "committés" une fois le stockage des objets effectué sans échec et l'indexation des métadonnées effectuée sans échec, avant notification au service versant

Chaque unité archivistique possède une et une seule entrée dans la collection LogbookLifeCycleUnit.

2.3.2 Exemple de JSON stocké en base

Extrait d'un JSON correspondant à un journal de cycle de vie d'une unité archivistique.

```
{
  "_id": "aeaqaaaaaehbl62nabqkwak3k7qg5tiaaaaq",
  "evId": "aedqaaaaaghbl62nabqkwak3k7qg5tiaaabq",
  "evType": "LFC.LFC_CREATION",
  "evDateTime": "2017-04-10T12:39:37.933",
  "evIdProc": "aedqaaaaaghe45hwabliwak3k7qg7kaaaaaq",
  "evTypeProc": "INGEST",
  "outcome": "STARTED",
  "outDetail": "LFC.LFC_CREATION.STARTED",
  "outMessg": "!LFC.LFC_CREATION.STARTED!",
  "agId": "{\"Name\":\"vitam-iaas-app-02\",\"Role\":\"worker\",\"ServerId\":
↪1041627981,\"SiteId\":1,\"GlobalPlatformId\":236321613}",
  "obId": "aeaqaaaaaehbl62nabqkwak3k7qg5tiaaaaq",
```

```
  "evDetData": null,
  "events": [
      {
          "evId": "aedqaaaaaghbl62nabqkwak3k7qg5tiaaabq",
          "evType": "LFC.CHECK_MANIFEST",
          "evDateTime": "2017-04-10T12:39:37.953",
          "evIdProc": "aedqaaaaaghe45hwabliwak3k7qg7kaaaaaq",
          "evTypeProc": "INGEST",
          "outcome": "STARTED",
          "outDetail": "LFC.CHECK_MANIFEST.STARTED",
          "outMessg": "Début de la vérification de la cohérence du bordereau",
          "agId": "{\"Name\":\"vitam-iaas-app-02\",\"Role\":\"worker\",\"ServerId\":
↪1041627981,\"SiteId\":1,\"GlobalPlatformId\":236321613}",
          "obId": "aeaqaaaaaehbl62nabqkwak3k7qg5tiaaaaq",
          "evDetData": null,
          "_tenant": 1
      },
      {
          "evId": "aedqaaaaaghbl62nabqkwak3k7qg5tiaaabq",
          "evType": "LFC.CHECK_MANIFEST.LFC_CREATION",
          "evDateTime": "2017-04-10T12:39:37.953",
          "evIdProc": "aedqaaaaaghe45hwabliwak3k7qg7kaaaaaq",
          "evTypeProc": "INGEST",
          "outcome": "OK",
          "outDetail": "LFC.CHECK_MANIFEST.LFC_CREATION.OK",
          "outMessg": "Succès de la création du journal du cycle de vie",
          "agId": "{\"Name\":\"vitam-iaas-app-02\",\"Role\":\"worker\",\"ServerId\":
↪1041627981,\"SiteId\":1,\"GlobalPlatformId\":236321613}",
          "obId": "aeaqaaaaaehbl62nabqkwak3k7qg5tiaaaaq",
          "evDetData": null,
          "_tenant": 1
      },{

      [...]

      }
  ],
  "_tenant": 1,
  "_v": 0
}
```

2.3.3 Détail des champs du JSON stocké en base

"_id" : Identifiant donné par le système lors de l'initialisation du journal du cycle de vie. Il est constitué d'une chaîne de 36 caractères correspondant à un GUID. Cet identifiant constitue la clé primaire du journal du cycle de vie de l'unité archivistique. Il reprend la valeur du champ _id de la collection Unit.

Ce champ existe uniquement pour la structure incluante.

"evId" (event Identifier) : identifiant de l'événement. Il est constitué d'une chaîne de 36 caractères correspondant à un GUID. Il identifie l'événement de manière unique dans la base.

Ce champ existe pour les structures incluantes et incluses

"evType" (event Type) : nom de l'événement. Il s'agit d'une chaîne de caractères. La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code est stocké dans ce champ, la traduction se fait via un fichier properties (vitam-logbook-message-fr.properties)

Ce champ existe pour les structures incluantes et incluses

“evDateTime” (event DateTime) [date de l’événement.] Il s’agit d’une date au format ISO8601 AAAA-MM-JJ+”T”+hh :mm :ss :[3digits de millisecondes] `Exemple : "2016-08-17T08:26:04.227"` Ce champ est positionné par le client LogBook.

Ce champ existe pour les structures incluantes et incluses

“evIdProc” (event Identifier Process) : identifiant du processus. Il s’agit d’une chaîne de 36 caractères. Toutes les mêmes entrées du journal du cycle de vie partagent la même valeur, qui est celle du champ “_id”

Ce champ existe pour les structures incluantes et incluses

“evTypeProc” (event Type Process) : type de processus. Il s’agit d’une chaîne de caractères. Nom du processus qui effectue l’action, parmi une liste de processus possibles fixée. Cette liste est disponible en annexe.

Ce champ existe pour les structures incluantes et incluses

“outcome” : statut de l’événement. Il s’agit d’une chaîne de caractères devant correspondre à une valeur de la liste suivante :

- STARTED (début de l’événement)
- OK (Succès de l’événement)
- KO (Echec de l’événement)
- WARNING (Succès de l’événement comportant des alertes)
- FATAL (Erreur technique)

Ce champ existe pour les structures incluantes et incluses

“outDetail” (outcome Detail) : code correspondant à l’erreur. Il s’agit d’une chaîne de caractères. Il contient le code fin de l’événement, incluant le statut. La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code est stocké dans ce champ, la traduction se fait via le fichier properties (vitam-logbook-message-fr.properties)

Ce champ existe pour les structures incluantes et incluses

“outMessg” (outcomeDetailMessage) : détail de l’événement. Il s’agit d’une chaîne de caractères. C’est un message intelligible destiné à être lu par un être humain en tant que détail de l’événement. Traduction du code présent dans outDetail issue du fichier vitam-logbook-message-fr.properties.

Ce champ existe pour les structures incluantes et incluses

“agId” (agent Identifier) : identifiant de l’agent réalisant l’action. Il s’agit de plusieurs chaînes de caractères indiquant le nom, le rôle et le PID de l’agent. Ce champ est calculé par le journal à partir de ServerIdentifier. `Exemple : {\"name\":\"ingest-internal_1\",\"role\":\"ingest-internal\",\"pid\":425367}`

Ce champ existe pour les structures incluantes et incluses

“obId” (object Identifier) : identifiant Vitam de l’objet ou du lot d’objets auquel s’applique l’opération (lot correspondant à une

Ce champ existe pour les structures incluantes et incluses

“evDetData” (event Detail Data) : détails des données de l’événement. Donne plus de détail sur l’événement. Par exemple, l’historisation de métadonnées lors d’une modification se fait dans ce champ. Dans la structure incluse correspondant à cet événement, il contient un JSON composé du champ suivant :

- diff : contient la différence entre les métadonnées d’origine et les métadonnées modifiées. Chaîne de caractères.

Ce champ existe pour les structures incluantes et incluses

“events” : tableau de structure Pour la structure incluante, le tableau contient N structures incluses dans l’ordre des événements (date)

Ce champ existe uniquement pour la structure incluante

“_tenant” : identifiant du tenant Il s’agit d’un entier. *Ce champ existe pour les structures incluantes et incluses*

“_v” : version de l’objet décrit

Il s’agit d’un entier.

Ce champ existe uniquement pour la structure incluante.

2.3.4 Détail des champs du JSON stocké en base spécifiques à une mise à jour

Exemple de données stockées :

```
"evDetData": "{\"diff\":\"-  Title : Recommandation de 2012 du CCSDS for Space Data␣
↪System Practices - Reference Model for an Open Archival Information System␣
↪(OAIS)\\n+  Title : Recommandation de 2012 du CCSDS for Space Data System Practices␣
↪- Reference Model for an Open Archival Information System (OAIS) 222\\n-
↪#operations : [ aedqaaaaacaam7mxabxecakz3jbfwpaaaaaq \\n+  #operations : [␣
↪aedqaaaaacaam7mxabxecakz3jbfwpaaaaaq, aecaaaaaacaam7mxabjssak2dzsjniyaaaaq \"}"
```

Dans le cas d'une mise à jour de métadonnées d'une unité archivistique (ArchiveUnit), le champ **"evDetData"** de l'événement final est composé des champs suivants :

"diff" : historisation des modifications de métadonnées. Son contenu doit respecter la forme suivante : les anciennes valeurs sont précédées d'un "-" (`-champ1: valeur1`) et les nouvelles valeurs sont précédées d'un "+" (`+champ1: valeur2`)

`Exemple : -Titre: Discours du Roi \n+Titre: Discours du Roi Louis XVI \n-Description: Etat Généraux du 5 mai 1789 \n+Description: Etat Généraux du 5 mai 1789 au Château de Versailles`

2.4 Collection LogbookLifeCycleObjectGroup

2.4.1 Utilisation de la collection LogbookLifeCycleObjectGroup

Le journal du cycle de vie du groupe d'objets (ObjectGroup) trace tous les événements qui impactent le groupe d'objets (et les objets associés) dès sa prise en charge dans le système. Il doit être conservé aussi longtemps que les objets sont gérés dans le système.

- dès la réception des objets, on trace les opérations effectuées sur les groupes d'objets et objets qui sont dans le SIP
- les journaux du cycle de vie sont "committés" une fois le stockage des objets effectué sans échec et l'indexation des métadonnées effectuée sans échec, avant notification au service versant

Chaque groupe d'objets possède une et une seule entrée dans sa collection LogbookLifeCycleObjectGroup.

2.4.2 Exemple de JSON stocké en base

```
{
  "_id": "aeaaaaaaaaaam7mxaap44akyf7hurgaaaaba",
  "evId": "aedqaaaaacaam7mxaap44akyf7hurgaaaabq",
  "evType": "CHECK_CONSISTENCY",
  "evDateTime": "2016-11-04T14:47:43.512",
  "evIdProc": "aedqaaaaacaam7mxaau56akyf7hr45qaaaaq",
  "evTypeProc": "INGEST",
  "outcome": "STARTED",
  "outDetail": "STARTED",
  "outMessg": "Début de la vérification de la cohérence entre objets/groupes d'objets␣
↪et ArchiveUnit.",
  "agId": "{\"Name\":\"vitam-iaas-worker-01\",\"Role\":\"worker\",\"PlatformId\":
↪425367}",
  "obId": "aeaaaaaaaaaam7mxaap44akyf7hurgaaaaba",
  "evDetData": null,
  "events": [
```

```
      {
          "evId": "aedqaaaaacaam7mxaap44akyf7hurgaaaabq",
          "evType": "CHECK_CONSISTENCY",
          "evDateTime": "2016-11-04T14:47:43.515",
          "evIdProc": "aedqaaaaacaam7mxaau56akyf7hr45qaaaaq",
          "evTypeProc": "INGEST",
          "outcome": "OK",
          "outDetail": "OK",
          "outMessg": "Objet/groupe dobjet référencé par un ArchiveUnit.",
          "agId": "{\"Name\":\"vitam-iaas-worker-01\",\"Role\":\"worker\",\
↪"PlatformId\":425367}",
          "obId": "aeaaaaaaaaaam7mxaap44akyf7hurgaaaaba",
          "evDetData": null,
          "_tenant": 0
      },
      {
          "evId": "\"aeaaaaaaaaaam7mxaap44akyf7hurgaaaaba\"",
          "evType": "CHECK_DIGEST",
          "evDateTime": "2016-11-04T14:47:45.132",
          "evIdProc": "aedqaaaaacaam7mxaau56akyf7hr45qaaaaq",
          "evTypeProc": "INGEST",
          "outcome": "STARTED",
          "outDetail": "STARTED",
          "outMessg": "Début de la vérification de lempreinte.",
          "agId": "{\"Name\":\"vitam-iaas-worker-01\",\"Role\":\"worker\",\
↪"PlatformId\":425367}",
          "obId": "aeaaaaaaaaaam7mxaap44akyf7hurgaaaaba",
          "evDetData": "{\"MessageDigest\":\
↪"0f1de441a7d44a277e265eb741e748ea18c96a59c8c0385f938b9768a42e375716dfa3b20cc125905636
          5aa0d3541f6128389ad60c8effbdc63b94df9a2e02bb\",\"Algorithm\": \"SHA512\", \
↪"SystemMessageDigest\": \"SHA-512\", \"SystemAlgorithm\": \
↪"0f1de441a7d44a277e265eb741e748ea18c96a59c8c0385f938b9768a42e375716dfa3b20cc125905636
          5aa0d3541f6128389ad60c8effbdc63b94df9a2e02bb\"} ",
          "_tenant": 0
      },
      {

          [...]

      }
  ],
  "_tenant": 0,
  "_v": 0
  }
```


2.4.3 Détail des champs du JSON stocké en base

"_id" : Identifiant donné par le système lors de l'initialisation du journal du cycle de vie. Il est constitué d'une chaîne de 36 caractères correspondant à un GUID. Il reprend la valeur du champ _id de la collection ObjectGroup. Cet identifiant constitue la clé primaire du journal du cycle de vie du groupe d'objet.

Ce champ existe uniquement pour la structure incluante.

"evId" (event Identifier) : identifiant de l'événement. Il est constitué d'une chaîne de 36 caractères correspondant à un GUID. Il identifie l'événement de manière unique dans la base.

Ce champ existe pour les structures incluantes et incluses

"evType" (event Type) : nom de l'événement. Il s'agit d'une chaîne de caractères. La liste des valeurs possibles

pour ce champ se trouve en annexe. Seul le code doit être stocké dans ce champ, la traduction doit se faire via le fichier properties (vitam-logbook-message-fr.properties).

Ce champ existe pour les structures incluantes et incluses

"evDateTime" (event DateTime) : date de l'événement. Il s'agit d'une date au format ISO8601 AAAA-MM-JJ+"T"+hh :mm :ss :[3digits de millisecondes] `Exemple : "2016-08-17T08:26:04.227"`. Ce champ est positionné par le client LogBook.

Ce champ existe pour les structures incluantes et incluses

"evIdProc" (event Identifier Process) : identifiant du processus. Il s'agit d'une chaîne de 36 caractères. Toutes les mêmes entrées du journal du cycle de vie partagent la même valeur, qui est celle du champ "_id".

Ce champ existe pour les structures incluantes et incluses

"evTypeProc" (event Type Process) : type de processus. Il s'agit d'une chaîne de caractères. Nom du processus qui effectue l'action, parmi une liste de processus possibles fixée. Cette liste est disponible en annexe.

Ce champ existe pour les structures incluantes et incluses

"outcome" : statut de l'événement. Il s'agit d'une chaîne de caractères devant correspondre une valeur de la liste suivante :

- STARTED (Début de l'événement)
- OK (Succès de l'événement)
- KO (Echec de l'événement)
- WARNING (Succès de l'événement comportant des alertes)
- FATAL (Erreur technique)

Ce champ existe pour les structures incluantes et incluses

"outDetail" (outcome Detail) : code correspondant à l'erreur Il s'agit d'une chaîne de caractères. Il contient le code fin de l'événement, incluant le statut. La liste des valeurs possibles pour ce champ se trouve en annexe. Seul le code est stocké dans ce champ, la traduction doit se faire via le fichier properties (vitam-logbook-message-fr.properties)

Ce champ existe pour les structures incluantes et incluses

"outMessg" (outcomeDetailMessage) : détail de l'événement. Il s'agit d'une chaîne de caractères. C'est un message intelligible destiné à être lu par un être humain en tant que détail de l'événement. Traduction du code présent dans outDetail issue du fichier vitam-logbook-message-fr.properties du code présent dans outDetail.

Ce champ existe pour les structures incluantes et incluses

"agId" (agent Identifier) : identifiant de l'agent réalisant l'action. Il s'agit de plusieurs chaînes de caractères indiquant le nom, le rôle et le PID de l'agent. Ce champ est calculé par le journal à partir de ServerIdentifier. `Exemple : {\"name\":\"ingest-internal_1\",\"role\":\"ingest-internal\",\"pid\":425367}`

Ce champ existe pour les structures incluantes et incluses

"obId" (object Identifier) : identifiant Vitam du lot d'objets auquel s'applique l'opération (lot correspondant à une liste).
Ce champ existe pour les structures incluantes et incluses

"evDetData" (event Detail Data) : détails des données de l'événement. Donne plus de détails sur l'événement. Par exemple, pour l'événement LFC.CHECK_DIGEST, lorsque l'empreinte d'un objet inscrite dans le bordereau n'est pas calculée en SHA512, ce champ précise l'empreinte d'origine et celle réalisée ensuite par la solution logicielle Vitam. Dans la structure incluse correspondant à cet événement, il contient un JSON composé des champs suivants :

- MessageDigest : empreinte de l'objet dans le bordereau. Chaîne de caractères. Reprends le champ "MessageDigest" du message ArchiveTransfer.
- Algorithm : algorithme de hachage utilisé dans le bordereau. Chaîne de caractères. Reprends l'attribut de champ "MessageDigest" du message ArchiveTransfer.
- SystemMessageDigest : empreinte de l'objet réalisé par la solution logicielle Vitam. Chaîne de caractères.
- SystemAlgorithm : algorithme de hachage utilisé par la solution logicielle Vitam. Chaîne de caractères.

Ce champ existe pour les structures incluantes et incluses

“events” : tableau de structure. Pour la structure incluante, le tableau contient N structures incluses dans l’ordre des événements (date)

Ce champ existe uniquement pour la structure incluante.

“_tenant” : identifiant du tenant. Il s’agit d’un entier.

Ce champ existe pour les structures incluantes et incluses

“_v” : version de l’objet décrit. Il s’agit d’un entier.

Ce champ existe uniquement pour la structure incluante.

CHAPITRE 3

Base MetaData

3.1 Collections contenues dans la base

La base Metadata contient les collections relatives aux métadonnées des unités archivistiques et des groupes d'objets.

3.2 Collection Unit

3.2.1 Utilisation de la collection Unit

La collection Unit contient les informations relatives aux unités archivistiques.

3.2.2 Exemple de JSON


```
{
   "_id": "aeaqaaaaaahbjs5eabbboak4educsrqaaaaq",
   "_og": "aebaaaaaaahh7uiuaakgeak46o5kxgaaaaaq",
   "DescriptionLevel": "RecordGrp",
   "Title": [
      "BAD0431E2C5E80E5BD42D547A3ED596444446.odt",
      "AU5"
   ],
   "Titles": {
      "fr": "BAD0431E2C5E80E5BD42D547A3ED596444446.odt"
   },
   "Description": "Maecenas sodales eleifend suscipit. Fusce vitae magna nec erat␣
↪bibendum imperdiet in vel eros.",
   "Descriptions": {
      "fr": "Maecenas sodales eleifend suscipit. Fusce vitae magna nec erat bibendum␣
↪imperdiet in vel eros."
   },
   "TransactedDate": "2016-06-03T15:28:00",
   "_sps": [
      "FRAN_NP_009913"
   ],
   "_sp": "FRAN_NP_009913",
   "_mgt": {
      "StorageRule": [
         {
```

```
            "Rule": "R1",
            "StartDate": "2017-05-01",
            "FinalAction": "RestrictAccess",
            "EndDate": "2018-05-01"
         }
      ],
      "OriginatingAgency": "FRAN_NP_009913"
   },
   "_ops": [
      "aedqaaaaachpuaosabkcgak4educq4yaaaaq"
   ],
   "_unitType": "INGEST",
   "_v": 0,
   "_tenant": 0,
   "_max": 3,
   "_min": 1,
   "_up": [
      "aeaqaaaaaahh7uiuaakgeak46o3locaaaaaq"
   ],
   "_nbc": 1,
   "_us": [
      "aeaqaaaaaahh7uiuaakgeak46o3ln6qaaaba",
      "aeaqaaaaaahh7uiuaakgeak46o3lodaaaaaq",
      "aeaqaaaaaahh7uiuaakgeak46o3locaaaaaq"
   ],
   "_uds": [
      {
         "aeaqaaaaaahh7uiuaakgeak46o3ln6qaaaba": 2
      },
      {
         "aeaqaaaaaahh7uiuaakgeak46o3lodaaaaaq": 2
      },
      {
         "aeaqaaaaaahh7uiuaakgeak46o3locaaaaaq": 1
      }
   ]
}
```


3.2.3 Exemple de XML en entrée

Ci-après, la portion d'un bordereau (manifest.xml) utilisée pour contribuer les champs du JSON. Il s'agit des informations situées entre les balises <ArchiveUnit>.

```
<?xml version="1.0" encoding="UTF-8"?>
<ArchiveUnit id="AU5">
   <Management>
      <StorageRule>
         <Rule>R1</Rule>
         <StartDate>2017-05-01</StartDate>
         <FinalAction>RestrictAccess</FinalAction>
      </StorageRule>
   </Management>
   <Content>
      <DescriptionLevel>RecordGrp</DescriptionLevel>
      <Title>AU5</Title>
   </Content>
```

```
   <ArchiveUnit id="ref3">
      <ArchiveUnitRefId>AU3</ArchiveUnitRefId>
   </ArchiveUnit>
</ArchiveUnit>
```

3.2.4 Détail du JSON

La structure de la collection Unit est composée de la transposition JSON de toutes les balises XML contenues dans la balise <DescriptiveMetadata> du bordereau conforme au standard SEDA v.2.0., c'est-à-dire toutes les balises se rapportant aux unités archivistiques.

Cette transposition se fait comme suit :

"_id" : identifiant unique de l'unité archivistique. Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

"_og" (objectGroup) : identifiant du groupe d'objets référencé dans cette unité archivistique. Il s'agit d'une chaîne de 36 caractères correspondant au GUID du champ _id de la collection objectGroup.

"_sps" : services producteurs liés à l'unité archivistique. Il s'agit d'un tableau contenant tous les services producteurs référençant l'unité archivistique. Il s'agit d'un tableau de chaînes de caractères.

"_sp" : service producteur d'origine. Il s'agit du service producteur inscrit dans le bordereau lié au transfert de l'unité archivistique. Il s'agit d'une chaîne de caractères.

"DescriptionLevel" : niveau de description archivistique de l'unité archivistique. Il s'agit d'une chaîne de caractères. Ce champ est renseigné avec les valeurs situées entre les balises <DescriptionLevel> dans le bordereau.

"Title" : titre de l'unité archivistique. Il s'agit d'une chaîne de caractères. Ce champ est renseigné avec les valeurs situées entre les balises <Title> dans le bordereau.

"Titles" : titres de l'unité archivistique par langue. Il s'agit d'un JSON. Les titres sont organisés sous la forme de clef - valeur, la clef étant l'indicatif de la langue, la valeur le titre. Par exemple : "fr" : "Maecenas sodales eleifend suscipit. Fusce vitae magna nec erat bibendum imperdiet in vel eros."

"Description" : description de l'unité archivistique. Il s'agit d'une chaîne de caractères. Ce champ est renseigné avec les informations situées entre les balises <description> de l'unité archivistique concernée dans le bordereau.

"Description" : description de l'unité archivistique par langue. Il s'agit d'un JSON. Les descriptions sont organisées sous la forme de clef - valeur, la clef étant l'indicatif de la langue, la valeur la description. Par exemple : "fr" : "Maecenas sodales eleifend suscipit. Fusce vitae magna nec erat bibendum imperdiet in vel eros."

"XXXXX" : des champs facultatifs peuvent être contenus dans le JSON lorsqu'ils sont renseignés dans le bordereau au niveau d
Se reporter à la documentation descriptive du SEDA 2.0 et notamment le schéma ontology.xsd pour connaître la liste des métadonnées facultatives)

"_ops" (operations) : tableau contenant les identifiants d'opérations auxquelles cette unité archivistique a participé.
Il s'agit d'une chaîne de 36 caractères correspondant au GUID du champs _id de la collection logBookOpération.

"_unitType" : champ indiquant le type d'unité archivistique concerné. Il s'agit d'une chaîne de caractères. La valeur contenue d

- INGEST : unité d'archivistique issue d'un SIP
- FILING_UNIT : unité d'archivistique issue d'un plan de classement
- HOLDING_UNIT : unité d'archivistique issue d'un arbre de positionnement

"_v" : version de l'objet décrit. Il s'agit d'un entier.

"_tenant" (tenant) : identifiant du tenant. Il s'agit d'un entier.

"_max" : profondeur maximale de l'unité archivistique par rapport à une racine. Calculée, cette profondeur correspond au maximum des profondeurs, quelles que soient les racines concernées et les chemins possibles.

"_min" : profondeur minimum de l'unité archivistique par rapport à une racine. Calculée, symétriquement le minimum des profondeurs, quel que soient les racines concernées et les chemins possibles.

"_up" : tableau recenssant les _id des unités archivistiques parentes (parents immédiats). Il s'agit d'une chaîne de 36 caractères correspondant au GUID. Valeur du champ _id de la collection Unit.

"_nbc" [nombre d'enfants immédiats de l'unité archivistique.] Il s'agit d'une chaîne de 36 caractères.

"_us" : tableau contenant la parentalité, indexé de la manière suivante [[GUID1, GUID2, ...].] Tableau de chaînes de 36 caractères.

"_uds" : tableau contenant la parentalité ainsi que le niveau de profondeur relative. Ces informations sont réunies dans le tableau sous la forme de clef/valeur. Exemple [{GUID1 : depth1}, {GUID2 : depth2}, ... }]. Il s'agit d'un tableau de JSON.

_profil : Type de document utilisé lors de l'entrée. Correspond à ArchiveUnitProfile, le profil d'archivage utilisé lors de l'entrée. Chaîne de caractères.

"_mgt" : contient les balises reprises du bloc <Management> du bordereau pour cette unité archivistique :

- "OriginatingAgency" : service producteur déclaré dans le message ArchiveTransfer (OriginatingAgencyIdentifier)
- "RuleType" [] : règles de gestion appliquées à cette unité archivistiques. Chaque tableau correspond à une catégorie de règle. Pour être valide, la catégorie de règle doit être présente dans la collection FileRules. Chaque tableau, optionnel, contient une à n règles. Chaque règle est composée des champs suivants : * "Rule" : identifiant de la règle. Pour être valide, elle doit être contenue dans la collection FileRules, et correspondre à la valeur du champ RuleID de la collection FileRules. * "StartDate" : date de début du calcul de l'échéance. Cette date est déclarée dans le message ArchiveTransfert ou ajoutée *a posteriori* par une modification. * "FinalAction" : champ décrivant le sort final. Ce champ est disponible pour les règles de catégorie "StorageRule" et "AppraisalRule". La valeur contenue dans le champ doit être disponible soit dans l'énumération FinalActionAppraisalCodeType soit dans FinalActionStorageCodeType. * "EndDate" : date de fin d'application de la règle ; Cette valeur est issue d'un calcul réalisé par la solution logicielle Vitam consistant en l'ajout du délai correspondant à la règle dans la collection FileRules et le champ startDate.

3.3 Collection ObjectGroup

3.3.1 Utilisation de la collection ObjectGroup

La collection ObjectGroup contient les informations relatives aux groupes d'objets.

3.3.2 Exemple de Json stocké en base


```
{
  "_id": "aebaaaaaaahbjs5eabbboak4d7shg4aaaaba",
  "_tenant": 0,
  "_profil": "",
  "FileInfo": {
      "Filename": "Filename0",
      "CreatingApplicationName": "CreatingApplicationName0",
      "CreatingApplicationVersion": "CreatingApplicationVersion0",
      "DateCreatedByApplication": "2006-05-04T18:13:51.0",
      "CreatingOs": "CreatingOs0",
      "CreatingOsVersion": "CreatingOsVersion0",
      "LastModified": "2006-05-04T18:13:51.0"
  },
  "_qualifiers": [{
```

```
    "qualifier": "PhysicalMaster",
        "_nbc": 1,
        "versions": [
            {
                "_id": "aeaaaaaaaahbjs5eabbboak4d7shg7iaaaaq",
                "DataObjectGroupId": "aebaaaaaaahbjs5eabbboak4d7shg4aaaaba",
                "DataObjectVersion": "PhysicalMaster_1",
                "PhysicalId": 123456789,
                "PhysicalDimensions": {
                    "Width": {
                        "unit": "centimetre",
                        "value": 1.7
                    },
                    "Height": {
                        "unit": "centimetre",
                        "value": 21
                    },
                    "Diameter": {
                        "unit": "centimetre",
                        "value": 22
                    },
                    "Length": {
                        "unit": "centimetre",
                        "value": 29.7
                    },
                    "Thickness": {
                        "unit": "centimetre",
                        "value": 1.4
                    },
                    "Weight": {
                        "unit": "kilogram",
                        "value": 1
                    },
                    "NumberOfPage": 20
                }
            }
        ]
    },
    {
        "qualifier": "BinaryMaster",
        "_nbc": 1,
        "versions": [
            {
                "_id": "aeaaaaaaaahbjs5eabbboak4d7shg4aaaaaq",
                "DataObjectGroupId": "aebaaaaaaahbjs5eabbboak4d7shg4aaaaba",
                "DataObjectVersion": "BinaryMaster_1",
                "FormatIdentification": {
                    "FormatLitteral": "Acrobat PDF 1.4 - Portable Document Format",
                    "MimeType": "application/pdf",
                    "FormatId": "fmt/18"
                },
                "FileInfo": {
                    "Filename": "Filename0",
                    "CreatingApplicationName": "CreatingApplicationName0",
                    "CreatingApplicationVersion": "CreatingApplicationVersion0",
                    "DateCreatedByApplication": "2006-05-04T18:13:51.0",
                    "CreatingOs": "CreatingOs0",
                    "CreatingOsVersion": "CreatingOsVersion0",
```

```
                    "LastModified": "2006-05-04T18:13:51.0"
                },
                "Size": 29403,
                "Uri": "Content/5zC1uD6CvaYDipUhETOyUWVEbxHmE1.pdf",
                "MessageDigest":
↪"942bb63cc16bf5ca3ba7fabf40ce9be19c3185a36cd87ad17c63d6fad1aa29d4312d73f2d6a1ba1266
                c3a71fc4119dd476d2d776cf2ad2acd7a9a3dfa1f80dc7",
                "Algorithm": "SHA-512",
                "_storage": {
                    "_nbc": 2,
                    "offerIds": [
                        "vitam-iaas-app-03.int",
                        "vitam-iaas-app-02.int"
                    ],
                    "strategyId": "default"
                }
            }
        ]
    }
  ],
  "_up": [
      "aeaqaaaaaahbjs5eabbboak4d7shg7qaaaaq"
  ],
  "_nbc": 0,
  "_ops": [
      "aedqaaaaachpuaosabkcgak4d7shenaaaaaq"
  ],
  "OriginatingAgency": "FRAN_NP_050056",
  "_v": 0,
  "_sps": [
      "FRAN_NP_050056"
  ]
}
```

3.3.3 Exemple de XML

Ci-après, la portion d'un bordereau (manifest.xml) utilisée pour contribuer les champ du JSON

```
<BinaryDataObject id="ID8">
    <DataObjectGroupReferenceId>ID4</DataObjectGroupReferenceId>
    <DataObjectVersion>BinaryMaster_1</DataObjectVersion>
    <Uri>Content/ID8.txt</Uri>
    <MessageDigest algorithm="SHA-512">
↪8e393c3a82ce28f40235d0870ca5b574ed2c90d831a73cc6bf2fb653c060c7f094fae941dfade786c826
    f8b124f09f989c670592bf7a404825346f9b15d155af</MessageDigest>
    <Size>30</Size>
    <FormatIdentification>
        <FormatLitteral>Plain Text File</FormatLitteral>
        <MimeType>text/plain</MimeType>
        <FormatId>x-fmt/111</FormatId>
    </FormatIdentification>
    <FileInfo>
        <Filename>BinaryMaster.txt</Filename>
        <LastModified>2016-10-18T21:03:30.000+02:00</LastModified>
    </FileInfo>
</BinaryDataObject>
```

3.3.4 Détail des champs du JSON

"_id" : identifiant du groupe d'objet. Il s'agit d'une chaîne de 36 caractères correspondant à un GUID. Cet id est ensuite reporté dans chaque structure incluse

"_tenant" : identifiant du tenant. Il s'agit d'un entier.

"_profil" : typologie de document. Repris du nom de la balise présente dans le <Metadata> du <DataObjectPackage> du bordereau qui concerne le BinaryMaster. Attention, il s'agit d'une reprise de la balise et non pas des valeurs à l'intérieur. Les valeurs possibles pour ce champ sont : Audio, Document, Text, Image et Video. Des extensions seront possibles (Database, Plan3D, ...).

"FileInfo" : reprend le bloc FileInfo du BinaryMaster. L'objet de cette copie est de pouvoir conserver les informations initiales du premier BinaryMaster (version de création), au cas où cette version serait détruite (selon les règles de conservation), car ces informations ne sauraient être maintenues de manière garantie dans les futures versions.

"_qualifiers" : tableau de structures décrivant les objets inclus dans ce groupe d'objets. Il est composé comme suit :

- "qualifier" : usage de l'objet. Ceci correspond à la valeur contenue dans le champ <DataObjectVersion> du bordereau. Par exemple pour <DataObjectVersion>BinaryMaster_1</DataObjectVersion>. C'est la valeur "BinaryMaster" qui est reportée.
    - "nb" : nombre d'objets correspondant à cet usage.
    - **"versions" : tableau des objets par version (une version = une entrée dans le tableau). Ces informations sont**
        - "_id" : identifiant de l'objet. Il s'agit d'une chaîne de 36 caractères corresppondant à un GUID.
        - "DataObjectGroupId" : identifiant du groupe d'objets. Chaîne de 36 caractères.
        - "DataObjectVersion" : version de l'objet par rapport à son usage.

    Par exemple, si on a *binaryMaster* sur l'usage, on aura au moins un objet *binarymaster_1*. Ces champs sont renseignés avec les valeurs récupérées dans les balises <DataObjectVersion> du bordereau.
    - **"FormatIdentification" : Contient trois champs qui permettent d'identifier le format du fichier. Une vérifica**
        - "FormatLitteral" : nom du format. C'est une reprise de la valeur située entre les balises <FormatLitteral> du message ArchiveTransfer.
        - "MimeType" : type Mime. C'est une reprise de la valeur située entre les balises <MimeType> du message ArchiveTransfer ou des valeurs correspondant au format tel qu'identifié par la solution logicielle Vitam.
        - "FormatId" : PUID du format de l'objet. Il est défini par la solution logicielle Vitam à l'aide du référentiel PRONOM maintenu par The National Archives (UK) et correspondant à la valeur du champ PUID de la collection FileFormat.
    - **"FileInfo" : Contient les informations sur les fichiers.**
        - "Filename" : nom de l'objet
        - "CreatingApplicationName" : nom de l'application avec laquelle l'objet a été créé. Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer. *Ce champ est facultatif et n'est pas présent systématiquement*
        - "CreatingApplicationVersion" : numéro de version de l'application avec laquelle le document a été créé. Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer. *Ce champ est facultatif et n'est pas présent systématiquement*

        - "CreatingOs" : système d'exploitation avec lequel l'objet a été créé. Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer. *Ce champ est facultatif et n'est pas présent systématiquement*
        - "CreatingOsVersion" : Version du système d'exploitation avec lequel l'objet a été créé. Ce champ est renseigné avec la métadonnée correspondante portée par le message ArchiveTransfer. *Ce champ et facultatif est n'est pas présent systématiquement*
        - "LastModified" : date de dernière modification de l'objet au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"Ce champ est optionnel, et est renseigné avec la métadonnée correspondante portée par le fichier.
        - "Size" : taille de l'objet (en octets). Ce champ contient un nombre entier.
    - "OtherMetadata" : Ce champ est renseigné avec les valeurs contenues entre les balises <OtherMetadata>. Ceci correspond à une extension du schéma SEDA du message ArchiveTransfert.
    - "Uri" : localisation du fichier correspondant à l'objet dans le SIP. Chaîne de caractères
    - "MessageDigest" : empreinte du fichier correspondant à l'objet. La valeur est calculée par la solution logicielle Vitam. Chaîne de caractères
    - "Algorithm" : algorithme utilisé pour réaliser l'empreinte du fichier correspondant à l'objet. Chaîne de caractères
    - **"_storage" : Contient trois champs qui permettent d'identifier les offres de stockage.**
        - "strategyId" : Identifiant de la stratégie de stockage.
        - "offerIds" : Liste des offres de stockage pour une stratégie donnée
        - "_nbc" : Nombre d'offres.

"_up" (unitup) : tableau identifiant les unités archivistiques parentes Il s'agit d'un tableau de chaînes de 36 caractères correspondant à un GUID contenu à la valeur contenue dans le champ _id de la collection Unit.

"_nbc" (nbobjects) : nombre d'objets dans le groupe d'objet. Il s'agit d'un entier.

"_ops" (operations) : tableau des identifiants d'opérations auxquelles ce GOT a participé. Il s'agit d'un tableau de chaînes de 36 caractères correspondant à un GUID contenu à la valeur contenue dans le champ _id de la collection LogBookOperation.

"OriginatingAgency" : service producteur déclaré dans le message ArchiveTransfer (OriginatingAgencyIdentifier) Il s'agit d'une chaîne de caractères.

"_sps" : services producteurs liées au groupe d'objets Il s'agit d'un tableau contenant tous les services producteurs référençant le groupe d'objet. Il s'agit d'un tableau de chaînes de caractère.

"_v" : version de l'objet décrit Il s'agit d'un entier.

"_sps" : services producteurs liées au groupe d'objets Il s'agit d'un tableau contenant tous les services producteurs référençant le groupe d'objet. Il s'agit d'un tableau de chaînes de caractère.

CHAPITRE 4

Base MasterData

4.1 Collections contenues dans la base

La base Masterdata contient les collections relatives aux référentiels utilisés par la solution logicielle Vitam.

4.2 Collection FileFormat

4.2.1 Utilisation de la collection FileFormat

La collection FileFormat permet de référencer et décrire les différents formats de fichiers ainsi que leur description. La collection est initialisée à partir de l'import du fichier de signature PRONOM, mis à disposition par The National Archive (UK).

4.2.2 Exemple de JSON stocké en base


```
{
  "_id": "aeaaaaaaaahbl62nabduoak3jc2zqciaadiq",
  "CreatedDate": "2016-09-27T15:37:53",
  "VersionPronom": "88",
   "PUID": "fmt/961",
  "Version": "2",
  "Name": "Mobile eXtensible Music Format",
  "Extension": [
      "mxmf"
  ],
  "HasPriorityOverFileFormatID": [
      "fmt/714"
  ],
  "MIMEType": "audio/mobile-xmf",
  "Group": "",
  "Alert": false,
  "Comment": "",
  "_v": 0
}
```

4.2.3 Exemple de la description d'un format dans le XML d'entrée

Ci-après, la portion d'un fichier de signature (DROID_SignatureFile_VXX.xml) utilisée pour renseigner les champ du JSON

```
<FileFormat ID="105" MIMEType="application/msword" Name="Microsoft Word for Macintosh␣
↪Document" PUID="x-fmt/64" Version="4.0">
  <InternalSignatureID>486</InternalSignatureID>
  <Extension>mcw</Extension>
</FileFormat>
```

4.2.4 Détail des champs du JSON stocké en base

"_id" : identifiant unique du format dans la solution logicielle Vitam. Il s'agit d'une chaîne de caractères composée de 36 caractères correspondant à un GUID.

"CreatedDate" : date de création de la version du fichier de signatures PRONOM utilisé pour initialiser la collection. Il s'agit d'une date au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"VersionPronom" : numéro de version du fichier de signatures PRONOM utilisé. Il s'agit d'un entier. Le numéro de version de PRONOM est à l'origine déclaré dans le fichier de signature au niveau de la balise <FFSignatureFile> au niveau de l'attribut "version ".

Dans cet exemple, le numéro de version est 88 :

```
<FFSignatureFile DateCreated="2016-09-27T15:37:53" Version="88" xmlns="http://www.
↪nationalarchives.gov.uk/pronom/SignatureFile">
```

"MIMEType" : MIMEtype correspondant au format de fichier. Il s'agit d'une chaîne de caractères. Elle est renseignée avec le contenu de l'attribut "MIMEType" de la balise <FileFormat>. Cet attribut est facultatif dans le fichier de signature.

"HasPriorityOverFileFormatID" : liste des PUID des formats sur lesquels le format a la priorité.

```
<HasPriorityOverFileFormatID>1121</HasPriorityOverFileFormatID>
```

Cet ID est ensuite utilisé dans Vitam pour retrouver le PUID correspondant. S'il existe plusieurs balises <HasPriorityOverFileFormatID> dans le fichier xml initial pour un format donné, alors les PUID seront stocké dans le JSON sous la forme suivante :

```
"HasPriorityOverFileFormatID": [
    "fmt/714",
    "fmt/715",
    "fmt/716"
],
```

"PUID" : identifiant unique du format au sein du référentiel PRONOM. Il s'agit d'une chaîne de caractères. Il est issu du champ "PUID" de la balise <FileFormat>. La valeur est composée du préfixe "fmt" ou "x-fmt", puis d'un nombre correspondant au numéro d'entrée du format dans le référentiel PRONOM. Les deux éléments sont séparés par un "/"

Par exemple

```
x-fmt/64
```

Les PUID comportant un préfixe “x-fmt” indiquent que ces formats sont en cours de validation par The National Archives (UK). Ceux possédant un préfixe “fmt” sont validés.

“Version” : version du format. Il s’agit d’une chaîne de caractères.

Exemples de formats :

```
Version="3D Binary Little Endian 2.0"
Version="2013"
Version="1.5"
```

L’attribut “version” n’est pas obligatoire dans la balise <fileformat> du fichier de signature.

“Name” : nom du format. Il s’agit d’une chaîne de caractères. Le nom du format est issu de la valeur de l’attribut “Name” de la balise <FileFormat> du fichier de signature.

“Extension” : Extension(s) du format. Il s’agit d’un tableau de chaînes de caractères. Il contient les valeurs situées entre les balises <Extension> elles-mêmes encapsulées entre les balises <FileFormat>. Le champ <Extension> peut-être multivalué. Dans ce cas, les différentes valeurs situées entre les différentes balises <Extensions> sont placées dans le tableau et séparées par une virgule.

Par exemple, pour le format dont le PUID est : fmt/918 on la XML suivant :

```
<FileFormat ID="1723" Name="AmiraMesh" PUID="fmt/918" Version="3D ASCII 2.0">
    <InternalSignatureID>1268</InternalSignatureID>
    <Extension>am</Extension>
    <Extension>amiramesh</Extension>
    <Extension>hx</Extension>
  </FileFormat>
```

Les valeurs des balises <Extensions> seront stockées de la façon suivante dans le JSON :

```
"Extension": [
     "am",
     "amiramesh",
     "hx"
 ],
```

“Alert” : alerte sur l’obsolescence du format. Il s’agit d’un booléen dont la valeur est par défaut placée à false.

“Comment” : commentaire. Il s’agit d’une chaîne de caractères. C’est un champ propre à la solution logicielle Vitam.

“Group” : Champ permettant d’indiquer le nom d’une famille de format.

Il s’agit d’une chaîne de caractères.

C’est un champ propre à la solution logicielle Vitam.

“_v” : version de l’objet décrit Il s’agit d’un entier.

4.3 Collection FileRules

4.3.1 Utilisation de la collection FileRules

La collection FileRules permet de stocker unitairement les différentes règles de gestion utilisées dans la solution logicielle Vitam pour calculer les échéances associées aux unités archivistiques.

Cette collection est alimentée par l’import d’un fichier CSV contenant l’ensemble des règles.

4.3.2 Exemple de JSON stocké en base

```
{
  "_id": "aeaaaaaaaahbl62nabduoak3jc4avsyaaaha",
  "_tenant": 0,
  "RuleId": "ACC-00011",
  "RuleType": "AccessRule",
  "RuleValue": "Communicabilité des informations portant atteinte au secret de la␣
↪défense nationale",
  "RuleDescription": "Durée de communicabilité applicable aux informations portant␣
↪atteinte au secret de la défense nationale\nL'échéance est calculée à partir de la␣
↪date du document ou du document le plus récent inclus dans le dossier",
  "RuleDuration": "50",
  "RuleMeasurement": "YEAR",
  "CreationDate": "2017-04-07",
  "UpdateDate": "2017-04-07",
  "_v": 0
 }
```

4.3.3 Structure du fichier d'import

RuleId	RuleType	RuleValue	RuleDescription	RuleDuration	RuleMeasurement
Id de la règle	Type de règle	Intitulé de la règle	Description de la règle	Durée	Unité de mesure de la durée

La liste des type de règles disponibles est en annexe 5.4.

Les valeurs renseignées dans la colonne unité de mesure doivent correspondre à une valeur de l'énumération RuleMeasurement

- MONTH
- DAY
- YEAR
- SECOND

4.3.4 Détail des champs

"_id" : identifiant unique par tenant de la règle de gestion. Il s'agit d'une chaîne de caractères composée de 36 caractères correspondant à une GUID.

"RuleId" : identifiant unique par tenant de la règle dans le référentiel utilisé. Il s'agit d'une chaîne de caractères. La valeur est reprise du champs RuleId du fichier d'import. Par commodité, les exemples sont composés d'un préfixe puis d'une nombre, séparés par un tiret, mais ce formalisme n'est pas obligatoire.

Par exemple :

```
ACC-00027
```

Les préfixes indiquent le type de règle dont il s'agit. La liste des valeurs pouvant être utilisées comme préfixes ainsi que les types de règles auxquelles elles font référence sont disponibles en annexe.

"RuleType" : *Champ obligatoire* Type de règle. Il s'agit d'une chaîne de caractères. Il correspond à la valeur située dans la colonne RuleType du fichier d'import. Les valeurs possibles pour ce champ sont indiquées en annexe.

"RuleValue" : *Champ obligatoire* Intitulé de la règle. Il s'agit d'une chaîne de caractères. Elle correspond à la valeur de la colonne RuleValue du fichier d'import.

“RuleDescription” : description de la règle. Il s’agit d’une chaîne de caractères. Elle correspond à la valeur de la colonne RuleDescriptionRule du fichier d’import.

“RuleDuration” : *Champ obligatoire* Durée de la règle. Il s’agit d’un entier compris entre 0 et 9999. Associé à la valeur indiqué dans RuleMeasurement, il permet de décrire la durée d’application de la règle de gestion. Il correspond à la valeur de la colonne RuleDuration du fichier d’import.

“RuleMeasurement” : *Champ obligatoire* Unité de mesure de la durée décrite dans la colonne RuleDuration du fichier d’import.

Il s’agit d’une chaîne de caractères devant correspondre à une valeur de l’énumération RuleMeasurementEnum, à savoir

- MONTH
- DAY
- YEAR
- SECOND

“CreationDate” : date de création de la règle dans la collection FileRule. Il s’agit d’une date au format ISO8601 AAAA-MM-JJ+”T”+hh :mm :ss :[3digits de millisecondes] `Exemple : "2016-08-17T08:26:04.227"`

“UpdateDate” : Date de dernière mise à jour de la règle dans la collection FileRules. Il s’agit d’une date au format ISO8601 AAAA-MM-JJ+”T”+hh :mm :ss :[3digits de millisecondes] `Exemple : "2016-08-17T08:26:04.227"`

“_v” : version de l’objet décrit. Il s’agit d’un entier.

4.4 Collection IngestContract

4.4.1 Utilisation de la collection

La collection IngestContract permet de référencer et décrire unitairement les contrats d’entrée.

4.4.2 Exemple de JSON stocké en base


```
{
  "_id": "aefqaaaaaahbl62nabkzgak3k6qtf3aaaaaq",
  "_tenant": 0,
  "Name": "SIA archives nationales",
  "Identifier": "IC-000012",
  "Description": "Contrat d'accès - SIA archives nationales",
  "Status": "ACTIVE",
  "CreationDate": "2017-04-10T11:30:33.798",
  "LastUpdate": "2017-04-10T11:30:33.798",
  "ActivationDate": "2017-04-10T11:30:33.798",
  "DeactivationDate": null,
  "ArchiveProfiles": [
      "ArchiveProfile8"
  ],
  "FilingParentId": "aeaqaaaaaagbcaacaax56ak35rpo6zqaaaaq",
  "_v": 0
}
```

4.4.3 Exemple d'un fichier d'import de contrat

Les contrats d'entrée sont importés dans la solution logicielle Vitam sous la forme d'un fichier JSON.

```
[
    {
        "Name":"Contrat Archives Départementales",
        "Description":"Test entrée - Contrat Archives Départementales",
        "Status" : "ACTIVE",
    },
    {
        "Name":"Contrat Archives Nationales",
        "Description":"Test entrée - Contrat Archives Nationales",
        "Status" : "INACTIVE",
        "ArchiveProfiles": [
          "ArchiveProfile8"
        ],
        "FilingParentId" : "aeaqaaaaaahkwxukabcg2ak4u2qq7eaaaaaq"
    }
]
```

Les champs à renseigner obligatoirement à la création d'un contrat sont :

- Name
- Description

Un fichier d'import peut décrire plusieurs contrats.

4.4.4 Détail des champs

"_id" : identifiant unique par tenant. Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

"_tenant" : information sur le tenant. Il s'agit de l'identifiant du tenant.

"Name" : *Champ obligatoire* Nom du contrat d'entrée, unique par tenant. Il s'agit d'une chaîne de caractères.

"Identifier" : *Champ obligatoire* Identifiant signifiant donné au contrat. Il est constitué du préfixe "IC-" suivi d'une suite de 6 chiffres. Par exemple : IC-007485. Il s'agit d'une chaîne de caractères.

"Description" : description du contrat d'entrée. Il s'agit d'une chaîne de caractères.

"Status" : statut du contrat. Peut être ACTIVE ou INACTIVE

"CreationDate" : date de création du contrat. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"LastUpdate" : date de dernière mise à jour du contrat dans la collection IngestContract. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"ActivationDate" : date d'activation du contrat. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"DeactivationDate" : date de désactivation du contrat. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"ArchiveProfiles" : liste des profils d'archivage pouvant être utilisés par le contrat d'entrée. Tableau de chaînes de caractères correspondant à la valeur du champ Identifier de la collection Profile.

"FilingParentId" : point de rattachement automatique des SIP en application de ce contrat correspondant à l'id d'une unité arc Il s'agit d'une chaîne de 36 caractères correspondant à un GUID dans le champ _id de la collection Unit.

"_v" : version de l'objet décrit Il s'agit d'un entier.

4.5 Collection AccessContract

4.5.1 Utilisation de la collection

La collection AccessContract permet de référencer et de décrire unitairement les contrats d'accès.

4.5.2 Exemple de JSON stocké en base


```
{
"_id": "aefqaaaaaahbl62nabkzgak3k6qtf3aaaaaq",
"_tenant": 0,
"Name": "SIA archives nationales",
"Identifier": "AC-000009",
"Description": "Contrat d'accès - SIA archives nationales",
"Status": "ACTIVE",
"CreationDate": "2017-04-10T11:30:33.798",
"LastUpdate": "2017-04-10T11:30:33.798",
"ActivationDate": "2017-04-10T11:30:33.798",
"DeactivationDate": null,
"OriginatingAgencies":["FRA-56","FRA-47"],
"DataObjectVersion": ["PhysicalMaster", "BinaryMaster", "Dissemination", "Thumbnail",
↪"TextContent"],
"WritingPermission": true,
"EveryOriginatingAgency": false,
"EveryDataObjectVersion": true,
"_v": 0
}
```


4.5.3 Exemple d'un fichier d'import de contrat d'accès

Les contrats d'entrée sont importés dans la solution logicielle Vitam sous la forme d'un fichier Json.

```
[
    {
        "Name":"Archives du Doubs",
        "Description":"Accès Archives du Doubs",
        "Status" : "ACTIVE",
        "ActivationDate":"10/12/2016",
        "OriginatingAgencies":["FRA-56","FRA-47"]
    },
    {
        "Name":"Archives du Calvados",
        "Description":"Accès Archives du Calvados",
        "Status" : "ACTIVE",
        "ActivationDate":"10/12/2016",
        "DeactivationDate":"10/12/2016",
        "OriginatingAgencies":["FRA-54","FRA-64"]
    }
]
```

Les champs à renseigner obligatoirement à la création d'un contrat sont :

- Name
- Description

Un fichier d'import peut décrire plusieurs contrats.

4.5.4 Détail des champs

"_id" : identifiant unique par tenant. Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

"_tenant" : information sur le tenant. Il s'agit de l'identifiant du tenant.

"Name" : *Champ obligatoire* Nom du contrat d'entrée unique par tenant. Il s'agit d'une chaîne de caractères.

"Identifier" : identifiant signifiant donné au contrat. Il est consituté du préfixe "AC-" suivi d'une suite de 6 chiffres. Par exemple : AC-001223. Il s'agit d'une chaîne de caractères.

"Description" : *Champ obligatoire* Description du contrat d'accès. Il s'agit d'une chaîne de caractères.

"Status" : statut du contrat. Peut être ACTIVE ou INACTIVE

"CreationDate" : date de création du contrat. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"LastUpdate" : date de dernière mise à jour du contrat dans la collection AccesContrat. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"ActivationDate" : date d'activation du contrat. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"DeactivationDate" : date de désactivation du contrat. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"OriginatingAgencies" : services producteurs pour lesquels le détenteur du contrat peut consulter les archives. Il s'agit d'un tableau de chaînes de caractères.

"DataObjectVersion" : usages d'un groupe d'objet auxquels le détenteur d'un contrat a access. Il s'agit d'un tableau de chaînes de caractères.

"WritingPermission" : droit d'écriture. Peut être true ou false. S'il est true, le détenteur du contrat peut effectuer des mises à jour.

"EveryOriginatingAgency" : droit de consultation sur tous les services producteurs. Il s'agit d'un booléen. Si la valeur est à true, alors le détenteur du contrat peut accéder aux archives de tous les services producteurs.

"EveryDataObjectVersion" : droit de consultation sur tous les usages. Il s'agit d'un booléen. Si la valeur est à true, alors le détenteur du contrat peut accéder à tous les types d'usages.

"_v" : version de l'objet décrit Il s'agit d'un entier.

4.6 Collection Profile

4.6.1 Utilisation de la collection

La collection Profile permet de référencer et décrire unitairement les profils SEDA.

4.6.2 Exemple de JSON stocké en base

```
{
  "_id": "aegaaaaaaehlfs7waax4iak4f52mzriaaaaq",
  "_tenant": 1,
  "Identifier": "PR-000003",
  "Name": "ArchiveProfile0",
  "Description": "aDescription of the Profile",
```

```
  "Status": "ACTIVE",
  "Format": "XSD",
  "CreationDate": "2016-12-10T00:00",
  "LastUpdate": "2017-05-22T09:23:33.637",
  "ActivationDate": "2016-12-10T00:00",
  "DeactivationDate": "2016-12-10T00:00",
  "_v": 1,
  "Path": "1_profile_aegaaaaaaehlfs7waax4iak4f52mzriaaaaq_20170522_092333.xsd"
}
```

4.6.3 Exemple d'un fichier d'import de profils

Un fichier d'import peut décrire plusieurs profils.

```
[
  {
    "Name":"ArchiveProfile0",
    "Description":"Description of the Profile",
    "Status":"ACTIVE",
    "Format":"XSD"
  },
    {
    "Name":"ArchiveProfile1",
    "Description":"Description of the profile 2",
    "Status":"ACTIVE",
    "Format":"RNG"
  }
]
```

Les champs à renseigner obligatoirement à la création d'un contrat sont :

- Name
- Description
- Format

4.6.4 Détail des champs

"_id" : identifiant unique. Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

"_tenant" : identifiant du tenant. Il s'agit d'un entier.

"Identifier" : Indique l'identifiant signifiant du profil SEDA. Il est consituté du préfixe "PR-" suivi d'une suite de 6 chiffres. Par exemple : PR-001573. Il s'agit d'une chaîne de caractères.

"Name" : *Champ obligatoire* Indique le nom du profil SEDA. Il s'agit d'une chaîne de caractères unique par tenant.

"Description" : *Champ obligatoire* Description du profil SEDA. Il s'agit d'une chaîne de caractères.

"Status" : Indique l'état du profil SEDA. Il s'agit d'une chaîne de caractères devant correspondre à une valeur de l'énuméartion ProfileStatus, soit ACTIVE soit INACTIVE.

"Format" : *Champ obligatoire* Indique le format attendu pour le fichier décrivant les règles du profil d'archivage. Il s'agit d'une chaîne de caractères devant correspondre à l'énumération ProfileFormat.

"CreationDate" : date de création du profil SEDA. Il s'agit d'une date au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

“LastUpdate” : date de dernière modification du profil SEDA dans la collection profile.. Il s’agit d’une date au format ISO 8601 YYY-MM-DD + ‘T’ + hh :mm :ss.millisecondes “+” timezone hh :mm. Exemple : “2016-08-19T16 :36 :07.942+02 :00”

“ActivationDate” : date d’activation du profil SEDA. Il s’agit d’une date au format ISO 8601 YYY-MM-DD + ‘T’ + hh :mm :ss.millisecondes “+” timezone hh :mm. Exemple : “2016-08-19T16 :36 :07.942+02 :00”

“DeactivationDate” : date de désactivation du profil SEDA. Il s’agit d’une date au format ISO 8601 YYY-MM-DD + ‘T’ + hh :mm :ss.millisecondes “+” timezone hh :mm. Exemple : “2016-08-19T16 :36 :07.942+02 :00”

“_v” : version de l’objet décrit Il s’agit d’un entier.

“Path” : Indique le chemin pour accéder au fichier du profil d’archivage. Chaîne de caractères.

4.7 Collection Context

4.7.1 Utilisation de la collection

La collection Context permet de stocker unitairement les contextes applicatifs

4.7.2 Exemple de JSON stocké en base

```
{
    "_id": "aegqaaaaaahkwxukabjosak4rp3kqkaaaaaq",
    "Name": "Contexte pour application 1",
    "Status": true,
    "Permissions": [
        {
            "_tenant": 1,
            "AccessContracts": [
                "AC-000017",
                "AC-000060"
            ],
            "IngestContracts": [
                "IC-000060"
            ]
        },
        {
            "_tenant": 0,
            "AccessContracts": [],
            "IngestContracts": []
        }
    ],
    "Identifier": "CT-000001"
}
```

Il est possible de mettre plusieurs contextes dans un même fichier, sur le même modèle que les contrats d’entrées ou d’accès par exemple. On pourra noter que le contexte est multi-tenant et définit chaque tenant de manière indépendante.

Les champs à renseigner obligatoirement à la création d’un contexte sont :

- Name
- Permissions. La valeur de Permissions peut cependant être vide : “Permissions : []”

4.7.3 Détail des champs

"_id" : identifiant unique dans l'ensemble du système. Il s'agit d'une chaîne de 36 caractères,.

"Name" : *Champ obligatoire* nom du contexte, qui doit être unique sur la plateforme. Il s'agit d'une chaîne de caractères.

"Identifier" : identifiant signifiant donné au contexte. Il est consituté du préfixe "CT-" suivi d'une suite de 6 chiffres. Par exemple : CT-001573. Il s'agit d'une chaîne de caractères.

"Status" : statut du contexte. Il peut être "true" ou "false" et a la valeur par défaut : "false". Selon son statut :

- "true" : le contexte est actif
- "false" : le contexte est inactif

"Permissions" : *Champ obligatoire* Début du bloc appliquant les permissions à chaque tenant. C'est un mot clé qui n'a pas de valeur associée.

"_tenant" : information sur le tenant. Il s'agit de l'identifiant du tenant dans lequel vont s'appliquer des permissions.

"AccessContracts" : tableau d'identifiants de contrats d'accès appliqués sur le tenant.

"IngestContracts" : tableau d'identifiants de contrats d'entrées appliqués sur le tenant.

"_v" : version de l'objet décrit Il s'agit d'un entier.

4.8 Collection AccessionRegisterSummary

4.8.1 Utilisation de la collection

Cette collection contient une vue macroscopique des fonds pris en charge dans la solution logicielle Vitam.

4.8.2 Exemple de JSON stocké en base


```
{
    "_id": "aefaaaaaaahkkoiuabp4sak3mmoj5vaaaaaq",
    "_tenant": 0,
    "OriginatingAgency": "Vitam",
    "TotalObjects": {
        "total": 27,
        "deleted": 0,
        "remained": 27
    },
    "TotalObjectGroups": {
        "total": 27,
        "deleted": 0,
        "remained": 27
    },
    "TotalUnits": {
        "total": 57,
        "deleted": 0,
        "remained": 57
    },
    "ObjectSize": {
        "total": 18292981,
```

```
        "deleted": 0,
        "remained": 18292981
    },
    "creationDate": "2017-04-12T17:01:11.764",
    "_v": 1
}
```

4.8.3 Exemple de la description dans le XML d'entrée

Les seuls élements issus du message ArchiveTransfer, utilisés ici sont ceux correspondants à la déclaration des identifiants du service producteur et du service versant. Ils sont placés dans le bloc <ManagementMetadata>

```
<ManagementMetadata>
         <OriginatingAgencyIdentifier>FRAN_NP_051314</OriginatingAgencyIdentifier>
         <SubmissionAgencyIdentifier>FRAN_NP_005761</SubmissionAgencyIdentifier>
</ManagementMetadata>
```

4.8.4 Détail des champs

"_id" : identifiant unique. Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

"_tenant" : identifiant du tenant. Il s'agit d'un entier.

"OriginatingAgency" : la valeur de ce champ est une chaîne de caractère. Ce champ est la clef primaire et sert de concaténation pour toutes les entrées effectuées sur ce producteur d'archives. Récupère la valeur contenue dans le bloc <OriginatinAgencyIdentifier> du message ArchiveTransfer.

Par exemple pour

```
<OriginatingAgencyIdentifier>FRAN_NP_051314</OriginatingAgencyIdentifier>
```

On récupère la valeur FRAN_NP_051314.

"TotalObjectGroups" : Contient la répartition du nombre de groupes d'objets du service producteur par état (total, deleted et remained)

- "total" : Nombre total de groupes d'objets pris en charge dans le système pour ce service producteur. La valeur contenue dans le champ est un entier.
- "deleted" : Nombre de groupes d'objets supprimés ou sortis du système. La valeur contenue dans ce champ est un entier.
- "remained" : Nombre actualisé de groupes d'objets conservés dans le système. La valeur contenue dans ce champ est un entier.

"TotalObjects" : Contient la répartition du nombre d'objets du service producteur par état (total, deleted et remained)

- "total" : Nombre total d'objets pris en charge dans le système pour ce service producteur. La valeur contenue dans le champ est un entier.
- "deleted" : Nombre d'objets supprimés ou sortis du système. La valeur contenue dans ce champ est un entier.
- "remained" : Nombre actualisé d'objets conservés dans le système. La valeur contenue dans ce champ est un entier.

"TotalUnits" : Contient la répartition du nombre d'unités archivistiques du service producteur par état (total, deleted et remained)

- "total" : Nombre total d'unités archivistiques pris en charge dans le système pour ce service producteur. La valeur contenue dans le champ est un entier.
- "deleted" : Nombre d'unités archivistiques supprimées ou sorties du système. La valeur contenue dans ce champ est un entier.
- "remained" : Nombre actualisé d'unités archivistiques conservées. La valeur contenue dans ce champ est un entier.

"ObjectSize" : Contient la répartition du volume total des fichiers du service producteur par état (total, deleted et remained)

- "total" : Volume total en octets des fichiers pris en charge dans le système pour ce service producteur. La valeur contenue dans le champ est un entier.
- "deleted" : Volume total en octets des fichiers supprimés ou sortis du système. La valeur contenue dans ce champ est un entier.
- "remained" : Volume actualisé en octets des fichiers conservés dans le système. La valeur contenue dans ce champ est un entier.

"creationDate" : Date d'inscription du producteur d'archives concerné dans le registre des fonds. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"_v" : Version de l'objet décrit Il s'agit d'un entier.

4.9 Collection AccessionRegisterDetail

4.9.1 Utilisation de la collection

Cette collection a pour vocation de référencer l'ensemble des informations sur les opérations d'entrées réalisées pour un service producteur. A ce jour, il y a autant d'enregistrements que d'opérations d'entrées effectuées pour ce service producteur, mais des évolutions sont d'ores et déjà prévues.

4.9.2 Exemple de JSON stocké en base


```
{
    "_id": "aedqaaaaakhpuaosabkcgak4ebd7deiaaaaq",
    "_tenant": 2,
    "OriginatingAgency": "FRAN_NP_009734",
    "SubmissionAgency": "FRAN_NP_009734",
    "ArchivalAgreement": "ArchivalAgreement0",
    "EndDate": "2017-05-19T12:36:52.572+02:00",
    "StartDate": "2017-05-19T12:36:52.572+02:00",
    "Status": "STORED_AND_COMPLETED",
    "LastUpdate": "2017-05-19T12:36:52.572+02:00",
    "TotalObjectGroups": {
        "total": 0,
        "deleted": 0,
        "remained": 0
    },
    "TotalUnits": {
        "total": 11,
        "deleted": 0,
        "remained": 11
    },
    "TotalObjects": {
```

```
        "total": 0,
        "deleted": 0,
        "remained": 0
    },
    "ObjectSize": {
        "total": 0,
        "deleted": 0,
        "remained": 0
    },
    "OperationIds": [
        "aedqaaaaakhpuaosabkcgak4ebd7deiaaaaq"
    ],
  "_v": 5
}
```


4.9.3 Exemple de la description dans le XML d'entrée

Les seuls élements issus du message ArchiveTransfer utilisés ici sont ceux correspondants à la déclaration des identifiants du service producteur et du service versant. Ils sont placés dans le bloc <ManagementMetadata>

```
<ManagementMetadata>
         <OriginatingAgencyIdentifier>FRAN_NP_051314</OriginatingAgencyIdentifier>
         <SubmissionAgencyIdentifier>FRAN_NP_005761</SubmissionAgencyIdentifier>
</ManagementMetadata>
```

4.9.4 Détail des champs

"_id" : identifiant unique. Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

"_tenant" : identifiant du tenant. Il s'agit d'un entier.

"OriginatingAgency" : Contient l'identifiant du service producteur. Il est issu du le bloc <OriginatinAgencyIdentifier>.

Par exemple :

```
<OriginatingAgencyIdentifier>FRAN_NP_051314</OriginatingAgencyIdentifier>
```

on récupère la valeur FRAN_NP_051314 La valeur est une chaîne de caractère.

"SubmissionAgency" : Contient l'identifiant du service versant. Il est contenu entre les baslises <SubmissionAgencyIdentifier>.

Par exemple pour

```
<SubmissionAgencyIdentifier>FRAN_NP_005761</SubmissionAgencyIdentifier>
```

On récupère la valeur FRAN_NP_005761. La valeur est une chaîne de caractères.

Ce champ est facultatif dans le bordereau. S'il' est absente ou vide, alors la valeur contenue dans le champ <OriginatingAgencyIdentifier> est reportée dans ce champ.

"ArchivalAgreement" : Contient le contrat utilisé pour réaliser l'entrée. Il est contenu entre les balises <ArchivalAgreement> et correspond à la valeur contenue dans le champ Name de la collection IngestContract.

Par exemple pour

```
<ArchivalAgreement>ArchivalAgreement0</ArchivalAgreement>
```

On récupère la valeur ArchivalAgreement0. La valeur est une chaîne de caractères.

"StartDate" : date de la première opération d'entrée correspondant à l'enregistrement concerné. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00".

"EndDate" : date de la dernière opération d'entrée correspondant à l'enregistrement concerné. La date est au format ISO 8601 YYY-MM-DD + 'T' + hh :mm :ss.millisecondes "+" timezone hh :mm. Exemple : "2016-08-19T16 :36 :07.942+02 :00"

"Status" : indication sur l'état des archives concernées par l'enregistrement. La liste des valeurs possibles pour ce champ se trouve en annexe 5.5.

"TotalObjectGroups" : Contient la répartition du nombre de groupes d'objets du fonds par état pour l'opération journalisée (to

- "total" : Nombre total de groupes d'objets pris en charge dans le cadre de l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- "deleted" : Nombre de groupes d'objets supprimés ou sortis du système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- "remained" : Nombre de groupes d'objets conservés dans le système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.

"TotalUnits" : Contient la répartition du nombre d'unités archivistiques du fonds par état pour l'opération journalisée (total, d

- "total" : Nombre total d'unités archivistiques pris en charge dans le cadre de l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- "deleted" : Nombre d'unités archivistiques supprimées ou sortis du système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- "remained" : Nombre d'unités archivistiques conservées dans le système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.

"TotalObjects" : Contient la répartition du nombre d'objets du fonds par état pour l'opération journalisée (total, deleted et rem

- "total" : Nombre total d'objets pris en charge dans le cadre de l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- "deleted" : Nombre d'objets supprimés ou sortis du système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- "remained" : Nombre d'objets conservés dans le système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.

"ObjectSize" : Contient la répartition du volume total des fichiers du fonds par état pour l'opération journalisée (total, deleted

- "total" : Volume total en octet des fichiers pris en charge dans le cadre de l'enregistrement concerné. La valeur contenue dans le champ est un entier.
- "deleted" : Volume total en octets des fichiers supprimés ou sortis du système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.
- "remained" : Volume total en octets des fichiers conservés dans le système pour l'enregistrement concerné. La valeur contenue dans ce champ est un entier.

4.10 Collection VitamSequence

4.10.1 Utilisation de collection

Cette collection permet de générer des identifiants signifiants pour les enregistrements des collections suivantes :

- IngestContract
- AccesContract
- Context
- Profil

Ces identifiants sont composés d'un préfixe de deux lettres, d'un tiret et d'une suite de six chiffres. Par exemple : IC-027593. Il sont reportés dans les champs Identifier des collections concernées.

4.10.2 Exemple de JSON stocké en base

```
{
  "_id": "aeaaaaaaaahkwxukabqteak4q5mtmdyaaaaq",
  "Name": "AC",
  "Counter": 44,
  "_tenant": 1,
  "_v": 0
}
```

4.10.3 Détail des champs

"_id" : Identifiant unique. Il s'agit d'une chaîne de 36 caractères correspondant à un GUID.

"Name" : préfixe. Il s'agit du préfixe utilisé pour générer un identifiant signifiant. La valeur contenue dans ce champ doit correspondre à la map du service VitamCounterService.java. La liste des valeurs possibles est détaillée en annexe 5.6. Il s'agit d'une chaîne de caractères.

"Counter" : numéro incrémental. Il s'agit du dernier numéro utilisé pour générer un identifiant signifiant. Il s'agit d'un entier.

"_tenant" : information sur le tenant Il s'agit de l'identifiant du tenant utilisant l'enregistrement Il s'agit d'un entier.

"_v" : version de l'objet décrit Il s'agit d'un entier.

CHAPITRE 5

Annexes

5.1 Valeurs possibles pour le champ evType du LogBook Operation

L'ensemble des étapes, tâches et traitements sont détaillés dans la documentation modèle de workflow

5.2 Valeurs possibles pour le champ evType du LogBook LifeCycle

L'ensemble des étapes, tâches et traitements sont détaillées dans la documentation modèle de workflow

5.3 Valeurs possibles pour le champ evTypeProc

Process Type	Valeur
Audit type process	AUDIT
Check type process	CHECK
Destruction type process (v2)	DESTRUCTION
Ingest type process	INGEST
Preservation type process	PRESERVATION
Rules Manager process	MASTERDATA
Traceability type process	TRACEABILITY
Update process	UPDATE

5.4 Catégories de règles possibles

Prefixe (Peut être modifié)	Type de règle correspondante	Description du type de règle
ACC	AccessRule	Règle d'accès / délai de communicabilité
APP	Appraisal	Règle correspondant à la durée d'utilité administrative (DUA)/ Durée de rétention
CLASS	ClassificationRule	Règle de classification
DIS	DisseminationRule	Règle de diffusion
REU	ReuseRule	Règle de réutilisation
STO	StorageRule	Durée d'utilité courante / durée de conservation au sens de la loi Informatique et Libertés

5.5 Valeurs possibles pour le champ Status de la collection AccessionRegisterDetail

Status type	Valeur
Le fonds est complet sauvegardé	STORED_AND_COMPLETED
Le fonds est mis à jour et sauvegardé	STORED_AND_UPDATED
Le fonds n'est pas sauvegardé	UNSTORED

5.6 Valeurs possibles pour le champ Name de la collection VitamSecquence

Prefixe	Type de collection correspondante	Decription
AC	AccessContract	Contrat d'accès
IC	IngestContract	Contrat d'entrée
PR	Profile	Profils
CT	Context	Contextes applicatifs