

VSN FIMS API

AUTHOR · Development Department

DATE · 11/05/2020

This document describes the REST FIMS API that will be used to integrate other applications with the VSNMAM (Media Asset Management) application.

v.1.2.

INDEX

1. Definitions, Acronyms and abbreviations
2. Introduction
 - 2.1 Purpose
 - 2.2 References
 - 2.3 Server's URLs
3. API FIMS. Particularities.
 - 3.1 XSD BMContent.
 - 3.2 XSD Repository.
 - 3.3 Common headers.
 - 3.3.1 Authorization
 - 3.3.2 Input and output format
 - 3.4 Working with JSON
4. VSNMAM API
 - 4.1 ContentQuery. (MAM/API/v1_2/query)
 - 4.1.1 Input message, ContentQueryRequest
 - 4.1.1.1 ContentQueryRequest Parameters.
 - 4.1.1.1.1 Query Profile.

4.1.1.1.2 QueryInputDefinition.

4.1.1.1.3 queryOutputDefinition

4.1.1.1.4 Additional Information QueryExpresion.

4.1.2 Output Message, ContentQueryResponse

4.2 Content. (MAM/API/v1_2/content/)

4.2.1 (GET) Obtain Content.

4.2.1.2 URL Parameters

4.2.1.2 Input message.

4.2.1.3 Output message. getContentResponse

4.2.2 (POST) Create content in system.

4.2.1.2 URL Parameters

4.2.1.2 Input message.

4.2.1.3 Output message. addContentResponse

4.3 Properties.(MAM/API/v1_2/content/properties)

4.3.1 URL Parameters

4.3.2 Input message. propertyInfos.

4.3.3 Output message. BMContentType

4.4 Review.(MAM/API/v1_2/review)

4.4.1 URL Parameters

4.4.2 Input message. propertyInfos.

4.4.3 Output message. SetPropertyReviewResponseType

4.5 Custom Public API. (MAM/ExtApi/v1_2_0_0/)

4.5.1 StorageNodes (StorageNodes/)

4.5.2 Volumes (Volumes/?storageNodeID={storageNodeID})

4.5.3 Folders

Folders/?storageNodeID={storageNodeID}&storageID={storageID}&volumeID={volumeID})

4.5. Essences (MAM/API/v1_2/essencePlaceholder)

4.5.1 Input message, AddEssencePlaceholderRequest

4.5.1.1 addEssencePlaceholderRequest Parameters.

4.5.1.1.1 essencePlaceholder. Extension attributes

4.5.2 Output message, AddEssencePlaceholderRequest

4.5.2.2 addEssencePlaceholderRequest Parameters.

4.5.2.2.1 essencePlaceholder. Extension attributes

4.6. Essences (MAM/API/v1_2/essence)

4.6.1 Input message, AddEssenceRequest

4.6.2 Output message, AddEssenceOperationAck

1. Definitions, Acronyms and abbreviations

Term	Definition
FIMS	Framework for Interoperable Media Services
AMWA	Advanced Media Workflow Association
EBU	European Broadcasting Union
SOA	Service Oriented Architecture
REST	Representational State Transfer
XSD	XML Schema Document
VSNMAM	VSN Media Asset Management

2. Introduction

2.1 Purpose

This document describes REST, XML and JSON messages that will be used with the interface of the application VSN Media Asset Management. These messages are used by the customers/clients to carry out searches, getting files, updating features/properties etc.

In the following sections XML/JSON will be showed in each operation and its response, within the messages we will remark the most important or those that are used by the system very often, note that most of the fields within the messages are optionals, the optional field can be dropped/left out by the customer even if they are shown is because of clarity.

2.2 References

<http://www.fims.tv/>

<https://github.com/fims-tv/fims>

2.3 Server's URLs

The server VSNMAM listens to messages in a web interface by mean of API REST, this API is compatible with 1.2 FIMS version.

The main API's are:

- Searches : Allows you to search for content.
 - https://vsnextplorer/MAM/API/v1_2/query
- Content: It allows to obtain a content of individual way through its resourceID.
 - https://vsnextplorer/MAM/API/v1_2/content/{resourceId}/{revisionID}?filter={filter}
- Update properties: Allows to perform property updates individually or in block.
 - https://vsnextplorer/MAM/API/v1_2/properties/{resourceId}/{revisionID}
- Essences: Allows you to manage files.
 - https://vsnextplorer/MAM/API/v1_2/essencePlaceholder
 - https://vsnextplorer/MAM/API/v1_2/essence

3. API FIMS. Particularities.

To start working with VSNMAM API, user has to take into account that the service waits messages in XML/JSON format following the steps given by FIMS.

It is mandatory to indicate that FIMS standar gives a handfull of extensions for the objects to let the scheme of each service that want to implement its standar can model data as they choose.

In case of VSNMAM, since metadata exists in each system linked to bussiness objects, VSN has decided to take advantage of that feature and extend the scheme using the working already establised on the FIMS standar.

This working consist on the fact that many ojects has a property named ExtensionGroup that lets the scheme of the object be extended the way we want, so it lets inject a XML. In our case this object is defined by bmContent, this feature has been used by VSN to include in all XML files the metadata that let define the VSNMAM sytem.

To know the schema in each customer we need to follow the following steps.

3.1 XSD BMContent.

In order to work with metadata defined in the system, and not only with the base of the XML within FIMS, is needed to first get the XSD that define that XML file, for that, each class of the VSNMAM system has its own metadata, once the identifier of the class is known, this information must be got from the VSNEXplorer system (VSNMAM service interface).

The screenshot shows a dark-themed interface with the following fields:

- Name ***: MultiFields
- Unique Identifier ***: MultiFields (highlighted in green)
- Name of the class in the API. It must be unique.*
- Type ***: 1

Once we have this unique identifier of the class we can carry out the following call to the system:

URL: <https://vsnextplorer/MAM/schemas/{ClassGuid}?version={VersionFIMS}>

Settings:

- **Class Guid:** It corresponds to the unique identifier that can be got from VSNExploere interface getting in the relevant class from what the schema must be validated.
- **Version FIMS:** Versión of FIMS, now we have 1.2.0 version or even is accepted to write 1_2_0 too, as we choose..

When executing this call will return a zip with all the .xsd necessary to validate the XML of the service calls that return bmContent, the XSD from which to depart would correspond with the name of the class.xsd

[Content_Types].xml	Documento XML
AssetMultiFields.xsd	XML Schema
baseMediaService-V1_2_0.xsd	XML Schema
baseTime-V1_2_0.xsd	XML Schema
description-V1_2_0.xsd	XML Schema
Id_a_borrar.xsd	XML Schema
vsnMamBase.xsd	XML Schema
xml.xsd	XML Schema

3.2 XSD Repository.

To work with API REST, objects defined in XSD and available on FIMS are used, so to see the XSD of these objects we need to follow the instructions we have in <http://www.fims.tv/>, we want to point out that in the following sections is explained the basic to carry out the calls available for API, though, we strongly recommend reviewing and becoming familiar with the standard.

3.3 Common headers.

In the HTTP calls it is mandatory to send a numbers of headers. These headers will be used to authenticate, input message format and output message format.

3.3.1 Authorization

All calls will need two headers that will let indicate on one hand the user and on the other hand the password to carry out the calls.

X-FIMS-UserName: *user*

X-FIMS-Password: *password*

3.3.2 Input and output format

The system lets work with XML and also with JSON, the headers to indicate what format we want to work with are within HTTP protocol standar.

Input format:

Content-Type: *application/xml*

Output format:

Accept: *application/xml*

3.4 Working with JSON

If you are going to carry out calls using xml or you have doubt about using JSON or XML, you can omit this section as long as you consider it.

We must take into account a series of conversion that are needed to get the whole information within a XML file is within a JSON file as well.

As you can see the most special or different to what we are accustomed are the nodes with the following names:

"@xmlns:rep": "<http://repository.fims.tv>" → Represents an attribute of xml, that in this case is a namespace.

Otherwise, if you look, the style would be similar to a regular json that defines an object..

```
"base:parameter": {
    "base:value": "test",
    "base:operator": "equal"
},
"base:name":
"bmContent.descriptions.description.bmContentDescription.identifier[0]",
```

4. VSNMAM API

The API that has the VSNMAM service, allows to search, obtain content, and update properties following the FIMS standard, as we have already mentioned, so for its use it is necessary to know at least the definition and use of these entries as it is defined in the standard documentation itself, so we urge you to become familiar with at least the repository part that defines this standard, if you want to understand a little more of the calls presented below.

4.1 ContentQuery. (MAM/API/v1_2/query)

This service entry allows you to search on the system, acting on the search engine, allows you to obtain any content that is in the system.

To carry out operations with this entry of the API it is necessary to take into account the following aspects:

Url: https://vsnextplorer/MAM/API/v1_2/query

Method HTTP: Post

Headers:

X-FIMS-UserName: user

X-FIMS-Password: password

Content-Type: *application/xml*

Accept: *application/xml*

4.1.1 Input message, ContentQueryRequest

(See FIMS documentation for more information).

<?xml version="1.0" encoding="utf-8"?>

<rep:contentQueryRequest

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:base="http://base.fims.tv"

xmlns:rep="http://repository.fims.tv">

<queryProfile>

version="1_2_0"

```

    <maxQueryResultItems>2</maxQueryResultItems>
    <maxQueryResultItemsPerPage>2</maxQueryResultItemsPerPage>
  </queryProfile>
  <queryInputDefinition>
    <pageNumber>0</pageNumber>
    <queryExpression>
<base:parameter>
    <base:name>bmContent.descriptions.description.bmContentDescription.identifier[0]</base:parameter>
    </base:name>
    <base:value>test</base:value>
    <base:operator>equal</base:operator>
    </base:parameter>
  </queryExpression>
</queryInputDefinition>
<queryOutputDefinition>
  <resultInfo>full</resultInfo>
  <resultSorting>
<sortFieldName>
    bmContent.descriptions.description.bmContentDescription.identifier[0]</sortFieldName>
    <sortDirection>ascending</sortDirection>
  </resultSorting>
</queryOutputDefinition>
</rep:contentQueryRequest>

```

4.1.1.1 ContentQueryRequest Parameters.

4.1.1.1.1 Query Profile.

maxQueryResultItemsPerPage → Indicates the maximum number of results that the page will return.

maxQueryResultItems → It has no effect at present, since it is a field that would restrict the search engine call to a certain number of elements, and we can not perform this operation currently, so it is not used or has any effect on the VSNMAM system.

4.1.1.1.2 QueryInputDefinition.

pageNumber → Starts at 0, indicates the page number.

QueryExpresion.

base:name → Indicates the parameter by which to perform the search. The following can be used:

`bmContent.descriptions.description.bmContentDescription.identifier[0]` → Asset.ID

`bmContent.descriptions.description.bmContentDescription.title[0]` → Asset.Title

`bmContent.descriptions.description.bmContentDescription.description[0]` → Asset.Description

`bmContent.resourceCreationDate` → Asset.CreationTime

`bmContent.resourceModifiedDate` → Asset.LastModificationTime

`bmContent.resourceID` → Asset.Guid

You can also search for variable metadata defined in the VSNMAM system, the addresses to use in this case all begin and have the following format.

`bmContent.descriptions.description.ExtensionGroup.tableName.columnName`

To know the table name and column name is necessary to have the schema, or an example of content that returns the routes to each property of the object.

To search by class or type of asset we can use the following addresses that correspond to an extension called VsnAssetStructure.

`bmContent.descriptions.description.ExtensionGroup.vsnAssetStructure.AssetClasses`

`bmContent.descriptions.description.ExtensionGroup.vsnAssetStructure.AssetType`

base:operator → The following are currently implemented::

equal

notEqual

greater

less

base:value → Indicates the value of the search parameter. You can use the * for text strings.

For entities and enumerated in the case of searching by variable metadata defined in the VSNMAM system, the value for equals or noequals must use the GUID of the entity, you must consult with the VSNMAM system administrator to obtain these values from the interface.

In the case of using vsnAssetStructure we have:

1. AssetClass : In the value you have to put the GUID of the class. Which can be obtained from the class file, as we have already mentioned.

2. AssetType: You have to set the integer that represents the enumerated, the possible values are those that are exposed next.

Media = 1

Bag = 2

Wedit = 3

Container = 4

NLEProject = 5

NLESequence = 6

Category = 7

4.1.1.1.3 queryOutputDefinition

resultInfo → The parameter is allowed, which will only return the number of results but the field contents will be empty.

noltems → It will only return the number of results but the field contents will be empty

light → It will only return the basic information of an asset. (BmContentDescription)

metadata→ It will also return the metadata information in an extension group.

physical→ It will return in addition to light, the information about files.

full→ It will return all possible information about the content.

Important comment: Both the physical and full parameters have a greater impact on the system, since it is necessary to obtain physical information from the files, so it is recommended to restrict the number of results to be obtained when it is going to be used, or to make a light call, and serialize the calls for the obtaining of the complete content by means of the entry of the API that allows to obtain a content of individual way from its resourceID.

resultSorting

sortFieldName → The same fields as in "base:name".

sortDirection → "Ascending" and "descending" are allowed that allow upward and downward respectively to sort the results.

Example queryOutputDefiniton:

```
<queryOutputDefinition>
  <resultInfo>full</resultInfo>
</resultSorting>

<sortFieldName>bmContent.descriptions.description.bmContentDescription.identifier[0]</sortFieldName>
  <sortDirection>asc</sortDirection>
</resultSorting>
</queryOutputDefinition>
```

4.1.1.1.4 Additional Information QueryExpresion.

The system supports the use of complex search expressions, using groups and rules that can be nested, using the parameter base: operator, which in addition to the values already specified can take the values of "and" and "or" for this aspect , We refer to an example where the definition of a group within another one of search rules is made. Also, for more information on this point we refer to the FIMS standard reference.

Example:

```

<queryExpression>
  <base:group>
    <base:parameters>
      <base:queryParameter>
        <base:name>bmContent.descriptions.description.bmContentDes
        cription.description[0]</base:name>
        <base:value>es*</base:value>
        <base:operator>equal</base:operator>
      </base:queryParameter>
      <base:queryParameter>
        <base:name>bmContent.resourceCreationDate</base:name>
        <base:value>2002-10-10T17:00:00Z</base:value>
        <base:operator>greater</base:operator>
      </base:queryParameter>
    </base:parameters>
  </base:group>
  <base:groups>
    <base:queryGroup>
      <base:parameters>
        <base:queryParameter>
          <base:name>bmContent.descriptions.description.bmCont
          entDescription.description[0]</base:name>
          <base:value>es*</base:value>
          <base:operator>notEqual</base:operator>
        </base:queryParameter>
      </base:parameters>
      <base:operator>and</base:operator>
    </base:queryGroup>
  </base:groups>
  <base:operator>and</base:operator>
</base:group>
</queryExpression>

```

4.1.2 Output Message, ContentQueryResponse

(See FIMS documentation for more information).

Example of response:

```

<?xml version="1.0" encoding="utf-8"?>
<contentQueryResponse version="1_2_0" xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <numberOfMatchingResults xmlns="">96101</numberOfMatchingResults>
  <pageNumber xmlns="">0</pageNumber>
  <request version="1_2_0" xmlns="">
    <credential>

```



```

    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <ExtensionAttributes xmlns="http://base.fims.tv"/>
  </credential>
  <queryProfile/>
  <queryInputDefinition>
    <pageNumber>0</pageNumber>
    <queryExpression>
      <parameter xmlns="http://base.fims.tv">

```

```

<name>bmContent.descriptions.description.bmContentDescription.identifier[0]</name>
  <value>*</value>
  <operator>equal</operator>
</parameter>
</queryExpression>
</queryInputDefinition>
<queryOutputDefinition>
  <resultInfo>full</resultInfo>
  <resultSorting>

```

```

<sortFieldName>bmContent.descriptions.description.bmContentDescription.identifier[0]
</sortFieldName>
  <sortDirection>ascending</sortDirection>
</resultSorting>
</queryOutputDefinition>
</request>
<contents xmlns="">
  <bmContent xmlns="http://base.fims.tv">
    <resourceID>01eba2c9-17ac-4472-b66a-51498b9524a4</resourceID>
    <resourceCreationDate>2016-04-27T15:23:21</resourceCreationDate>
    <resourceModifiedDate>2016-05-05T14:47:25</resourceModifiedDate>
    <ExtensionGroup/>
    <ExtensionAttributes/>
    <qaReportReferences/>
    <descriptions>
      <description>
        <resourceID>01eba2c9-17ac-4472-b66a-51498b9524a4</resourceID>
        <ExtensionGroup>
          <vsAssetStructure xmlns="http://vsn-tv.com/vsnMamBase">
            <AssetClass displayName="FIMS API">FIMS API</AssetClass>
            <AssetType displayName="Media">1</AssetType>
            <AssetCategory displayName="Rushes">

```

```

    <resourceID
xmlns="http://base.fims.tv">873833e1-4d22-46b9-8783-50121710c8f1</resourceID>
        </AssetCategory>
    </vsnAssetStructure>
    <MultivaluedTypes displayName="Tipo multivalor" set="FIMS.API.Tests"
xmlns="http://vsn-tv.com/FIMS.API.Tests">
    <AssetList displayName="Lista de contenidos">
        <bmContent xmlns="http://base.fims.tv">

<resourceID>61bebb91-73da-47af-a29b-25a631b8d3de</resourceID>
        </bmContent>
        <bmContent xmlns="http://base.fims.tv">

<resourceID>61bebb91-73da-47af-a29b-25a631b8d3de</resourceID>
        </bmContent>
    </AssetList>
        <EntityList displayName="Lista de entidades"
xmlns:vMam="http://vsn-tv.com/vsnMamBase">
        <Entity/>
    </EntityList>
    </MultivaluedTypes>
</ExtensionGroup>
<ExtensionAttributes/>
<bmContentDescription xmlns="http://description.fims.tv">
    <title>Assets duplicados</title>
    <creator>
        <contactDetails>

<username>S-1-431902345-579497581-937514854-1171364678-3863120297-789528
731</username>
        <occupation/>
        <gender/>
    </contactDetails>
    </creator>
    <description>test</description>
    <identifier>0285270416</identifier>
    <version/>
    </bmContentDescription>
</description>
</descriptions>
<status>

```

```

    <resourceID>01eba2c9-17ac-4472-b66a-51498b9524a4</resourceID>
    <ExtensionGroup/>
    <ExtensionAttributes/>
    <status>online</status>
  </status>
</bmContent>
  <resourceID>fc5409a8-fd59-4a9f-867c-689f7a92d22a</resourceID>
  <resourceCreationDate>2016-04-21T14:16:26</resourceCreationDate>
  <resourceModifiedDate>2016-04-21T14:17:37</resourceModifiedDate>
  <ExtensionGroup/>
  <ExtensionAttributes/>
  <qaReportReferences/>
  <descriptions>
    <description>
      <resourceID>fc5409a8-fd59-4a9f-867c-689f7a92d22a</resourceID>
      <ExtensionGroup>
        <vsAssetStructure xmlns="http://vsn-tv.com/vsnMamBase">
          <AssetClass displayName="CTC Files">ctc.file.version</AssetClass>
          <AssetType displayName="Media">1</AssetType>
        </vsAssetStructure>
        <CtcTraffic displayName="Traffic" set="CTC_MTD"
xmlns="http://vsn-tv.com/CTC_MTD">
          <ReadyToAir displayName="Ready to Air">false</ReadyToAir>
          <enumversiontype displayName="Version Type"/>
        </CtcTraffic>
      </ExtensionGroup>
      <ExtensionAttributes/>
      <bmContentDescription xmlns="http://description.fims.tv">
        <title>testDeleteAssetPistolet</title>
        <creator>
          <contactDetails>
            <username>S-1-431902345-579497581-937514854-1171364678-3863120297-789528
731</username>
            <occupation/>
            <gender/>
          </contactDetails>
        </creator>
        <description/>
        <identifier>CTCF0228210416CTCF</identifier>
        <version/>

```

```

    </bmContentDescription>
  </description>
</descriptions>
<status>
  <resourceID>fc5409a8-fd59-4a9f-867c-689f7a92d22a</resourceID>
  <ExtensionGroup/>
  <ExtensionAttributes/>
  <status>online</status>
</status>
</bmContent>
</contents>
</contentQueryResponse>

```

At the first level of the response object we can find:

numberOfMatchingResults As you can see the response object, it is composed of the number of results that match

pageNumber That indicates the page number returned.

request It must be a copy of the input message with which the request was made.

contents It has an array of bmContent objects that represent the assets that have coincided in the search, these objects are defined in FIMS, and are the basic object that returns the repository, and will contain in the ExtensionGroup the variable metadata of the VSNMAM system, as we have already commented, and that this object should independently follow the scheme or schemes referred to in this documentation previously.

4.2 Content. (MAM/API/v1_2/content/)

This service entry allows you to obtain/create a Content object from the system, from its resourceID.

4.2.1 (GET) Obtain Content.

To perform operations with this api entry it is necessary to take into account the following aspects:

Url: https://vsnextplorer/MAM/API/v1_2/content/{resourceId}/{revisionID}

Method HTTP: GET

Headers:

X-FIMS-UserName: user

X-FIMS-Password: password

Content-Type: *application/xml*

Accept: *application/xml*

4.2.1.2 URL Parameters

resourceId→ Id of the resource from which the content is to be obtained.

revisionID→ Not currently used, because the VSNMAM system does not currently support versions but is necessary to follow the call scheme defined in the FIMS standard, any value can be set. Ex: 1

filter→ Tipo BMContentFilterType. The default value is“full”.

BMContentFilterType Its possible values are:

full → Returns all available information.

min→ Only returns the minimum information, which includes the resourceId and asset modification and creation dates.

metadata→ Returns the metadata information related to the content.

physical→ Returns file information including physical paths.

specific→ Not currently used.

Call example:

https://vsnextplorer/MAM/API/v1_2/content/c8c26a96-120f-4d31-b6e3-bba2b224c0c8/1?filter=min

4.2.1.2 Input message.

There is no incoming message.

4.2.1.3 Output message. getContentResponse

```

<getContentResponse xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="1_2_0">
  <content xmlns="">
    <resourceID
xmlns="http://base.fims.tv">01eba2c9-17ac-4472-b66a-51498b9524a4</resourceID>
    <resourceCreationDate
xmlns="http://base.fims.tv">2016-04-27T15:23:21</resourceCreationDate>
    <resourceModifiedDate
xmlns="http://base.fims.tv">2016-05-05T14:47:25</resourceModifiedDate>
    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <ExtensionAttributes xmlns="http://base.fims.tv"/>
    <qaReportReferences xmlns="http://base.fims.tv"/>
    <descriptions xmlns="http://base.fims.tv">
      <description>
        <resourceID>01eba2c9-17ac-4472-b66a-51498b9524a4</resourceID>
        <ExtensionGroup>
          <vsAssetStructure xmlns="http://vsn-tv.com/vsnMamBase">
            <AssetClass displayName="PamCategory">PamCategory</AssetClass>
            <AssetType displayName="Category">7</AssetType>
            <AssetCategory displayName="Rushes">
              <resourceID
xmlns="http://base.fims.tv">873833e1-4d22-46b9-8783-50121710c8f1</resourceID>
            </AssetCategory>
          </vsAssetStructure>
        </ExtensionGroup>
        <ExtensionAttributes/>
        <bmContentDescription xmlns="http://description.fims.tv">
          <title>Assets duplicados</title>
          <creator>
            <contactDetails>
              <username>
                S-1-431902345-579497581-937514854-1171364678-3863120297-789528731
              </username>
              <occupation/>
              <gender/>
            </contactDetails>
          </creator>
          <description>test</description>
        </bmContentDescription>
      </description>
    </descriptions>
  </content>
</getContentResponse>

```

```

    <identifier>0285270416</identifier>
    <version/>
  </bmContentDescription>
</description>
</descriptions>
<status xmlns="http://base.fims.tv">
  <resourceID>01eba2c9-17ac-4472-b66a-51498b9524a4</resourceID>
  <ExtensionGroup/>
  <ExtensionAttributes/>
  <status>online</status>
</status>
</content>
</getContentResponse>

```

This message basically encapsulates a bmContent, so what has been said so far is valid in the same way. And the only different thing about the base objects bmContent of FIMS is its extensionGroup, which as we have already mentioned will contain the variable metadata of the VSMAM system, and that is defined in the scheme that is obtained from the own system.

As you could notice the yellow part of xml has the extensionGroup defined by VSN System, now you could see 3 parameters.

1. AssetClass which represents the class of an asset(bmContent).
2. AssetType which represents the type of an asset(bmContent).
3. AssetCategory which represents the category of an asset(bmContent), also this category is, in turn, an asset (bmContent), so the assets are in Production Asset Management will contain in the vsnAssetStructure a AssetCategory to represents this concept.

4.2.2 (POST) Create content in system.

To perform operations with this api entry it is necessary to take into account the following aspects:

Url: https://vsnexplorer/MAM/API/v1_2/content/

Method HTTP: POST

Headers:

X-FIMS-UserName: user

X-FIMS-Password: password

Content-Type: *application/xml*

Accept: *application/xml*

4.2.1.2 URL Parameters

This call allows create object from `addContentRequest` which it's defined in FIMS scheme in the repository part. Some valid request returns `addContentResponse` object.

Call example:

https://vsnextplorer/MAM/API/v1_2/content/

4.2.1.2 Input message.

```
<addContentRequest xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:schemaLocation="http://repository.fims.tv ../repository-V1_2_0.xsd" version="1_2_0">
  <content xmlns="">
    <resourceID xmlns="http://base.fims.tv"></resourceID>
    <descriptions xmlns="http://base.fims.tv">
      <description>
        <resourceID></resourceID>
        <ExtensionGroup>
          <vsAssetStructure xmlns="http://vsn-tv.com/vsnMamBase">
            <AssetClass>TodoEnUno</AssetClass>
            <AssetCategory displayName="Rushes">
              <resourceID xmlns="http://base.fims.tv">
                873833e1-4d22-46b9-8783-50121710c8f1
              </resourceID>
            </AssetCategory>
          </vsAssetStructure>
        </ExtensionGroup>
      </description>
    </descriptions>
  </content>
</addContentRequest>
```


It's necessary to modify AssetClass in vsnAssetStructure to indicate the class of asset you would need to create, this parameter indicates the GUID of the class you could obtain from the data of the asset template in VSNE Explorer.



You could create an asset category type or any other type defined in the asset [type](#), in the creation it is not necessary to specify the type because it is inherited from AssetClass.

If you would create an asset in some specific category in the Production Asset Management, you could specify the category with the parameter AssetCategory.

```
<AssetCategory displayName="Rushes">
  <resourceID xmlns="http://base.fims.tv">
    873833e1-4d22-46b9-8783-50121710c8f1
  </resourceID>
</AssetCategory>
```

The display name of AssetCategory is unused in the creation process, then you can avoid it.

Additional note:

When you will create categories, you must specify an AssetCategory, which category created it will inherit the related volumes and rights.

Now we can't create categories in the Production Asset Management (PAM), which are in the root category (first level), because we need to specify related volumes and rights, if you try to create a root category, you will receive a message indicating this restriction:

```
{"Guid":"fe80::35f2:a38:f1b0:788c%24:2367_d6b46804-959d-4b9e-b427-8f87f68e371e","Code":500,"Msg":"You could not create categories in root with API."}
```

Even you could add a concrete guid for asset if you put the guid in resourceID field in the first node of content.

You create an asset in the system with metadata for that, you could send an addContentRequest object with metadata, which it represents by ExtensionGroup tag, remember you need the schema of the asset that represents the metadata, this is explain in this [here](#).

This example shows you could added an asset with metatada and added in the Rushes category PAM:

```
<addContentRequest xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
version="1_2_0">
  <content xmlns="">
    <resourceID xmlns="http://base.fims.tv">
    </resourceID>
    <descriptions xmlns="http://base.fims.tv">
      <description>
        <resourceID>
        </resourceID>
        <ExtensionGroup>
          <vsnAssetStructure xmlns="http://vsn-tv.com/vsnMamBase">
            <AssetClass>TodoEnUno</AssetClass>
            <AssetCategory displayName="Rushes">
              <resourceID xmlns="http://base.fims.tv">
                873833e1-4d22-46b9-8783-50121710c8f1
              </resourceID>
            </AssetCategory>
          </vsnAssetStructure>
          <Tipos xmlns="http://vsn-tv.com/ContentSetTipos"
displayName="Tipo" set="ContentSetTipos">
            <Tip01 displayName="Tip01">
              <resourceID
xmlns="http://base.fims.tv">3c213b15-144e-4c75-a800-af592aed9a90
              </resourceID>
            </Tip01>
            <Tip02 displayName="Tip02">
              <bmContent xmlns="http://base.fims.tv">
```

```

<resourceID>3c213b15-144e-4c75-a800-af592aed9a90</resourceID>
  </bmContent>
  <bmContent xmlns="http://base.fims.tv">

<resourceID>7ca47f1c-a3be-401c-975d-6f096be9f7ed</resourceID>
  </bmContent>
  </Tipo2>
  <bool displayName="Boolean">>false</bool>
  <entidad xmlns:vMam="http://vsn-tv.com/vsnMamBase"
displayName="Entity" />
  <enum displayName="Enum" />
  <entidadlista xmlns:vMam="http://vsn-tv.com/vsnMamBase"
displayName="Entity List">
  <entitiesTipos />
  </entidadlista>
</Tipos>
</ExtensionGroup>
<bmContentDescription xmlns="http://description.fims.tv">
  <title>PAM Asset From FIMS Api</title>
  <description>Description of the asset</description>
  <identifier>Todo0002181115Mod</identifier>
</bmContentDescription>
</description>
</descriptions>
</content>
</addContentRequest>

```

4.2.1.3 Output message. addContentResponse

```

<addContentResponse version="1_2_0" xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <content xmlns="">
    <resourceID
xmlns="http://base.fims.tv">0b7bd87c-15c8-484d-9763-ea356f0b8b6a</resourceID
>
    <resourceCreationDate
xmlns="http://base.fims.tv">2019-09-26T15:04:56</resourceCreationDate>
    <resourceModifiedDate
xmlns="http://base.fims.tv">2019-09-26T15:04:56</resourceModifiedDate>
    <ExtensionGroup xmlns="http://base.fims.tv" />

```

```

<ExtensionAttributes xmlns="http://base.fims.tv" />
<qaReportReferences xmlns="http://base.fims.tv" />
<descriptions xmlns="http://base.fims.tv">
  <description>
    <resourceID>0b7bd87c-15c8-484d-9763-ea356f0b8b6a</resourceID>
    <ExtensionGroup>
      <vsAssetStructure xmlns="http://vsn-tv.com/vsnMamBase">
        <AssetClass displayName="PamCategory">TodoEnUno</AssetClass>
        <AssetType displayName="Category">1</AssetType>
        <AssetCategory displayName="Rushes">
          <resourceID xmlns="http://base.fims.tv">

```

873833e1-4d22-46b9-8783-50121710c8f1

```
</resourceID>
```

```
</AssetCategory>
```

```
</vsAssetStructure>
```

```
<Tipos xmlns="http://vsn-tv.com/ContentSetTipos"
displayName="Tipo" set="ContentSetTipos">
```

```
<Tipo1 displayName="Tipo1">
```

```
<resourceID xmlns="http://base.fims.tv">
```

3c213b15-144e-4c75-a800-af592aed9a90

```
</resourceID>
```

```
</Tpo1>
```

```
<Tpo2 displayName="Tpo2">
```

```
<bmContent xmlns="http://base.fims.tv">
```

```
<resourceID>3c213b15-144e-4c75-a800-af592aed9a90</resourceID>
```

```
</bmContent>
```

```
<bmContent xmlns="http://base.fims.tv">
```

```
<resourceID>7ca47f1c-a3be-401c-975d-6f096be9f7ed</resourceID>
```

```
</bmContent>
```

```
</Tpo2>
```

```
<bool displayName="Boolean">>false</bool>
```

```
</Tipos>
```

```
</ExtensionGroup>
```

```
<ExtensionAttributes />
```

```
<bmContentDescription xmlns="http://description.fims.tv">
```

```
<title>RaizDesdeFIMSCOnFormatoResource</title>
```

```
<creator>
```

```

<contactDetails>

<username>S-1-431902345-579497581-937514854-1171364678-3863120297-789528731<
/username>
    <occupation />
    <gender />
</contactDetails>
</creator>
<description />
<identifier>PamC0018190926PamC</identifier>
<version />
</bmContentDescription>
</description>
</descriptions>
<status xmlns="http://base.fims.tv">
    <resourceID>0b7bd87c-15c8-484d-9763-ea356f0b8b6a</resourceID>
    <ExtensionGroup />
    <ExtensionAttributes />
    <status>online</status>
</status>
</content>
</addContentResponse>

```

4.3 Properties.(MAM/API/v1_2/content/properties)

This service entry allows you to update one or more fields of a bmContent object of the system, from its resourceID.

To perform operations with this api entry it is necessary to take into account the following aspects:

Url: https://vsnextplorer/MAM/API/v1_2/content/properties/{resourceId}/{revisionID}

Method HTTP: PUT

Headers:

X-FIMS-UserName: *user*

X-FIMS-Password: *password*

Content-Type: *application/xml*

Accept: *application/xml*

4.3.1 URL Parameters

resourceId→ Id of the resource to be updated.

revisionID→ Not currently used, because the VSNMAM system does not currently support versions but is necessary to follow the call scheme defined in the FIMS standard, any value can be set, Ex:1.

4.3.2 Input message. propertyInfos.

```

<propertyInfos xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <propertyInfo>
    <address xmlns=""> bmContent.descriptions[0].bmContentDescription.title</address>
    <action xmlns="">update</action>
    <value xmlns=""> titulo asset </value>
  </propertyInfo>
  <propertyInfo>
    <address xmlns="">
bmContent.descriptions[0].bmContentDescription.description</address>
    <action xmlns="">update</action>
    <value xmlns=""> descripcion del asset</value>
  </propertyInfo>
  <propertyInfo>
    <address xmlns="">
bmContent.descriptions.description.ExtensionGroup.FichaCliente.ITV_Cliente</address>
    <action xmlns="">update</action>
    <value xmlns=""> guid de una entidad o enumerado </value>
  </propertyInfo>
</propertyInfos>

```

Address: It indicates the path to the value to be modified. You can modify the values of the field marked in orange following this section, since it follows the same rules and explanations given that for the base field: name

Action: Currently only the update is supported.

Value: Final value that you want to give the content data.

4.3.3 Output message. BMContentType

```
<?xml version="1.0" encoding="utf-8"?>
<BMContentType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://base.fims.tv">79a12dc2-1201-437e-a0d0-587176c069eb</resourceID
<resourceCreationDate
xmlns="http://base.fims.tv">2015-04-15T08:56:46</resourceCreationDate>
<resourceModifiedDate
xmlns="http://base.fims.tv">2016-07-08T07:49:00</resourceModifiedDate>
<ExtensionGroup xmlns="http://base.fims.tv"/>
<ExtensionAttributes xmlns="http://base.fims.tv"/>
<qaReportReferences xmlns="http://base.fims.tv"/>
<descriptions xmlns="http://base.fims.tv">
<description>
<resourceID>79a12dc2-1201-437e-a0d0-587176c069eb</resourceID>
<ExtensionGroup>
<vsAssetStructure xmlns="http://vsn-tv.com/vsnMamBase">
<AssetClass displayName="MultiFields">MultiFields</AssetClass>
<AssetType displayName="Media">1</AssetType>
</vsAssetStructure>
<todoslostiposdecampo displayName="Todos los tipos de campo"
set="Id_a_borrar" xmlns="http://vsn-tv.com/Id_a_borrar">
<campoentero displayName="Campo Booleano2">>false</campoentero>
<campodate displayName="Campo Date">2015-05-26Z</campodate>
<allboolean displayName="Boolean">>true</allboolean>
<alldate displayName="Date">2015-02-12Z</alldate>
<alldatetime displayName="DateTime">2015-01-02T00:00:00Z</alldatetime>
<allentity displayName="Entity"
xmlns:vMam="http://vsn-tv.com/vsnMamBase">
<vMam:id>528</vMam:id>
```

```

    <vMam:name>hhhheth</vMam:name>
  </allentity>
  <allenum displayName="Enum" name="hola" id="887" guid="1111"/>
  <allinteger displayName="Integer">2</allinteger>
  <alllongtext displayName="LongText">4</alllongtext>
  <allreal displayName="Real">1</allreal>
  <allshorttext displayName="Short Text">&lt;p&gt;cortosdfdsdfsdf&lt;/p&gt;
&lt;p&gt;&amp;nbsp;&lt;/p&gt;
&lt;p&gt;&lt;strong&gt;asd&lt;/strong&gt;&lt;/p&gt;</allshorttext>
  <alltime displayName="Time">18:15:07Z</alltime>
  </todoslostiposdecampo>
</ExtensionGroup>
<ExtensionAttributes/>
<bmContentDescription xmlns="http://description.fims.tv">
  <title>pruebajuant</title>
  <creator>
    <contactDetails>

<username>S-1-431902345-579497581-937514854-1171364678-3863120297-789528
731</username>
    <occupation/>
    <gender/>
  </contactDetails>
</creator>
  <description>descripción de asset</description>
  <identifier>Mult0004150415</identifier>
  <version/>
</bmContentDescription>
</description>
</descriptions>
<status xmlns="http://base.fims.tv">
  <resourceID>79a12dc2-1201-437e-a0d0-587176c069eb</resourceID>
  <ExtensionGroup/>
  <ExtensionAttributes/>
  <status>online</status>
</status>
</BMContentType>

```

This message basically encapsulates a bmContent of an asset, so what has been said so far for this object is valid in the same way. The only different thing about the bmContent base objects of FIMS is its extensionGroup, which as we have already mentioned will

contain the variable metadata of the VSMAM system, which is defined in the scheme that is obtained from the system itself.

4.4 Review.(MAM/API/v1_2/review)

This service entry allows you to set a review of one or more fields of a bmContent object of the system, from its resourceID.

To perform operations with this api entry it is necessary to take into account the following aspects:

Url: https://vsnextplorer/MAM/API/v1_2/review/{resourceId}/{revisionID}

Method HTTP: PUT

Headers:

X-FIMS-UserName: *user*

X-FIMS-Password: *password*

Content-Type: *application/xml*

Accept: *application/xml*

4.4.1 URL Parameters

resourceId → Id of the resource to be updated.

revisionID → Guid of the review to be created or updated.

4.4.2 Input message. propertyInfos.

```
<propertyInfos xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <propertyInfo>
    <address xmlns=""> bmContent.descriptions[0].bmContentDescription.title</address>
    <action xmlns="">update</action>
    <value xmlns=""> titulo asset </value>
  </propertyInfo>
  <propertyInfo>
    <address xmlns="">
bmContent.descriptions[0].bmContentDescription.description</address>
```

```

<action xmlns="">update</action>
<value xmlns=""> descripcion del asset</value>
</propertyInfo>
</propertyInfo>
<address xmlns="">
bmContent.descriptions.description.ExtensionGroup.FichaCliente.ITV_Cliente</address>
<action xmlns="">update</action>
<value xmlns=""> guid de una entidad o enumerado </value>
</propertyInfo>
</propertyInfos>

```

Address: It indicates the path to the value to create review. You can modify the values of the field marked in orange following this section, since it follows the same rules and explanations given that for the base field: name

Action: Currently only the update is supported.

Value: Final value that you want to give the content data review.

4.4.3 Output message. SetPropertyReviewResponseType

```

<?xml version="1.0" encoding="utf-8"?>
<setPropertiesReviewResponse xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <version>1_2_0</version>
  <revisionID>ae021186-4a75-45b6-a1dc-0f2b6d5b6034</revisionID>
</BMContentType>

```

This message basically encapsulates a setPropertiesReviewResponse, and this type is composed by two fields:

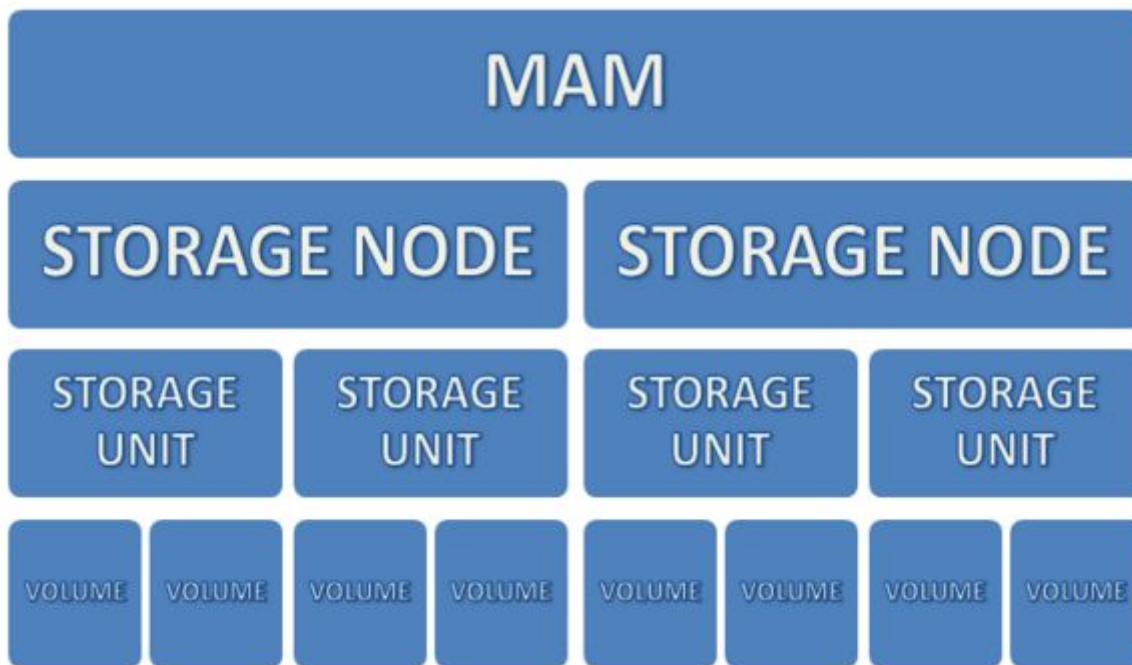
version Represents API version.

revisionID Is the guid of the created or updated review. If we want to update an existing review, we must use this field as revisionID in URL parameter.

4.5 Custom Public API. (MAM/ExtApi/v1_2_0_0/)

This resource groups all the calls necessary to work with the MAM service, but they do not have a place in the FIMS's own API. In the FIMS flow, we have a "knownLocations" object that represents locations known by the system, that allow us to leave a file and through the "AddEssence" calls have the system copy them, however in the current MAM flow, it is the user must choose where he wants to locate the file once integrated into the system, so it is necessary to pass certain data to the "AddEssencePlaceholder" call.

Why are these calls necessary? To work with files in the MAM service, it is necessary to use the calls that are exposed below to be able to choose a final destination for the file once it has been uploaded to the system.



4.5.1 StorageNodes (StorageNodes/)

```
<storageNodes xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://repository.vsn.tv">
  <storageNode>
    <id>1</id>
    <name>Testingserver2</name>
    <endPoint>https://testingserver2/storage</endPoint>
  </storageNode>
```

```

<storageNode>
  <id>3005</id>
  <name>TVV</name>
  <endPoint>https://TVV/storage</endPoint>
</storageNode>
</storageNodes>

```

4.5.2 Volumes (Volumes/?storageNodeID={storageNodeID})

```

<volumes xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://repository.vsn.tv">
  <volume>
    <id>1</id>
    <name>HiRes</name>
    <storageNodeId>1</storageNodeId>
    <storageId>1</storageId>
    <storageType>online</storageType>
    <accesses>
      <access>
        <id>2</id>
        <url>\\dc01\Production\HiRes</url>
        <type>unc</type>
      </access>
      <access>
        <id>12</id>
        <url>https://production/storage/Downloads/HiRes</url>
        <type>https</type>
      </access>
    </accesses>
    <hasFolders>true</hasFolders>
  </volume>
  <volume>
    <id>2</id>
    <name>Proxy</name>
    <storageNodeId>1</storageNodeId>
    <storageId>2</storageId>
    <storageType>online</storageType>
    <accesses>
      <access>
        <id>5</id>
        <url>\\production\LowRes</url>
        <type>unc</type>
      </access>
    </accesses>
  </volume>
</volumes>

```

```

</access>
<access>
  <id>6</id>
  <url>https://production/storage/Downloads/Proxy</url>
  <type>https</type>
</access>
<access>
  <id>7</id>
  <url>ump://192.168.100.215:2012/production/LowRes</url>
  <type>ump</type>
</access>
</accesses>
<hasFolders>>false</hasFolders>
</volume>
</volumes>

```

4.5.3 Folders (Folders/?storageNodeId={storageNodeId}&storageId={storageId}&volumeId={volumeId})

In case of storage type is Online the you could do request to /folders resources to obtain folder details.

```

<folders xmlns="http://repository.vsn.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <folder>
    <id>1</id>
    <name>Test</name>
    <parentId xsi:nil="true" />
    <storageNodeId>1</storageNodeId>
    <storageId>1</storageId>
    <volumeId>1</volumeId>
  </folder>
  <folder>
    <id>6</id>
    <name>CTC</name>
    <parentId>1</parentId>
    <storageNodeId>1</storageNodeId>
    <storageId>1</storageId>
    <volumeId>1</volumeId>
  </folder>
</folders>

```

```

<id>4</id>
<name>jbarcelo</name>
<parentId>1</parentId>
<storageNodeId>1</storageNodeId>
<storageId>1</storageId>
<volumeId>1</volumeId>
</folder>
<folder>
  <id>5</id>
  <name>FIMS uploads</name>
  <parentId>4</parentId>
  <storageNodeId>1</storageNodeId>
  <storageId>1</storageId>
  <volumeId>1</volumeId>
</folder>
<folder>
  <id>3</id>
  <name>KissTV</name>
  <parentId>1</parentId>
  <storageNodeId>1</storageNodeId>
  <storageId>1</storageId>
  <volumeId>1</volumeId>
</folder>
<folder>
  <id>2</id>
  <name>Video</name>
  <parentId xsi:nil="true" />
  <storageNodeId>1</storageNodeId>
  <storageId>1</storageId>
  <volumeId>1</volumeId>
</folder>
</folders>

```

4.5.Essences (MAM/API/v1_2/essencePlaceholder)

To work with these API requests, you should obtain several information with alternative API defined in 4.4 Custom public API. (MAM/ExtApi/v1_2_0_0/)

4.5.1 Input message, AddEssencePlaceholderRequest

(See FIMS documentation for more information).

The most important change is we use ExtensionAttributes defined in essencePlaceholder object provided in FIMS to send request with information about the placeholder in MAM.

```
<addEssencePlaceholderRequest version="1_2_0" xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <content xmlns="">
    <resourceID
xmlns="http://base.fims.tv">854B5CFE-8518-4157-93BF-E98F1A3243A4</resourceID>
    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <ExtensionAttributes xmlns="http://base.fims.tv"/>
    <qaReportReferences xmlns="http://base.fims.tv"/>
    <descriptions xmlns="http://base.fims.tv"/>
    <status xmlns="http://base.fims.tv">
      <ExtensionGroup/>
      <ExtensionAttributes/>
      <status>new</status>
    </status>
  </content>
  <format xmlns="">
    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <ExtensionAttributes xmlns="http://base.fims.tv"/>
    <bmEssenceLocators xmlns="http://base.fims.tv"/>
    <formatCollection xmlns="http://base.fims.tv"/>
    <duration xmlns="http://base.fims.tv"/>
    <contentType xmlns="http://base.fims.tv"/>
  </format>
  <essencePlaceholder xmlns="">
    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <!--746 UNC access-->
    <!--923 HTTPS access-->
    <ExtensionAttributes xmlns="http://base.fims.tv"
StorageNodeID="1" StorageID="546" VolumenID="548" FolderID="22"
AccessID="746" FilenameTarget="hola.mxf"/>
    <storageType xmlns="http://base.fims.tv">online</storageType>
    <containerMimeType xmlns="http://base.fims.tv"/>
    <status xmlns="http://base.fims.tv">
      <ExtensionGroup/>
      <ExtensionAttributes/>
      <status>new</status>
      <subStatus>
        <ExtensionGroup/>
        <ExtensionAttributes/>
      </subStatus>
    </status>
  </essencePlaceholder>
</addEssencePlaceholderRequest>
```

4.5.1.1 addEssencePlaceholderRequest Parameters.

4.5.1.1.1 essencePlaceholder. Extension attributes

StorageNodeID → You can obtain from custom api.

StorageID → You can obtain from custom api.

VolumenID → You can obtain from custom api.

FolderID → You can obtain from custom api.

AccessID → You can obtain from custom api.

FilenameTarget → It must be provided from original requester.

4.5.2 Output message, AddEssencePlaceholderRequest

```
<?xml version="1.0" encoding="utf-8"?>
<addEssencePlaceholderResponse version="1_2_0" xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <content xmlns="">
    <resourceID
xmlns="http://base.fims.tv">64b8a6a7-afb4-4be1-bcfa-71e653d27864</resourceID>
    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <ExtensionAttributes xmlns="http://base.fims.tv"/>
    <qaReportReferences xmlns="http://base.fims.tv"/>
    <descriptions xmlns="http://base.fims.tv"/>
    <status xmlns="http://base.fims.tv">
      <ExtensionGroup/>
      <ExtensionAttributes/>
      <status>new</status>
    </status>
  </content>
  <essencePlaceholderLocator xmlns="">
    <resourceCreationDate
xmlns="http://base.fims.tv">0001-01-01T00:00:00</resourceCreationDate>
    <resourceModifiedDate
xmlns="http://base.fims.tv">0001-01-01T00:00:00</resourceModifiedDate>
    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <ExtensionAttributes StorageNodeID="1" StorageID="1" VolumenID="1"
FolderID="1" AccessID="1" FilenameTarget="hola.mxf"
TargetInfo="https://testingserver2/StorageEx/api/File/upload/27537"
TargetPath="\\dc01\Testingserver2\Datos\Daily\Test\hola.mxf" TransferID="27537"
VSNFileGroupID="145659" xmlns="http://base.fims.tv"/>
    <storageType xmlns="http://base.fims.tv">online</storageType>
    <containerMimeType xmlns="http://base.fims.tv"/>
```



```

<status xmlns="http://base.fims.tv">
  <ExtensionGroup/>
  <ExtensionAttributes/>
  <status>new</status>
</status>
</essencePlaceholderLocator>
</addEssencePlaceholderResponse>

```

In response the user could see the information about where the path is, then you could copy the file into the path provided by targetPath attribute or upload file by http request into the TargetInfo.

When user or automatic system or program copy the file into the final destination you must call the next request AddEssence to finalize transfer and update to final status in MAM system.

4.5.2.2 addEssencePlaceholderRequest Parameters.

4.5.2.2.1 essencePlaceholder. Extension attributes

TargetInfo → Path where would upload file.

TargetPath → Path where would copy file.

TransferID → Internal ID representation transfer of MAM VSN System.

VSNFileGroupID → Internal ID representation file of MAM VSN System.

4.6.Essences (MAM/API/v1_2/essence)

This request allows finalize transfers in the MAM VSN System. When you finalize copy or upload a file, you must call the request with the correct information that you could obtain in addEssencePlaceholder, you could use directly the `essencePlaceholderLocator` object but be carefully you need change the object to `essenceplaceholder` when you fill AddEssenceRequest then MAM call to storage service to put the transfer status in OK. Then the system launch events to generate related stuff like Low-Res file, keyframes, etc.

4.6.1 Input message, AddEssenceRequest

```
<addEssenceRequest      version="1_2_0"      xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <content xmlns="">
    <resourceID
xmlns="http://base.fims.tv">854B5CFE-8518-4157-93BF-E98F1A3243A4</resourceID>
    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <ExtensionAttributes xmlns="http://base.fims.tv"/>
    <qaReportReferences xmlns="http://base.fims.tv"/>
    <descriptions xmlns="http://base.fims.tv"/>
    <status xmlns="http://base.fims.tv">
      <ExtensionGroup/>
      <ExtensionAttributes/>
      <status>new</status>
    </status>
  </content>
  <essencePlaceholder xmlns="">
    <resourceCreationDate
xmlns="http://base.fims.tv">0001-01-01T00:00:00</resourceCreationDate>
    <resourceModifiedDate
xmlns="http://base.fims.tv">0001-01-01T00:00:00</resourceModifiedDate>
    <ExtensionGroup xmlns="http://base.fims.tv"/>
    <ExtensionAttributes StorageNodeID="1" StorageID="546" VolumenID="548"
FolderID="22"      AccessID="746"      FilenameTarget="hola.mxf"
TargetInfo="https://testingserver2/Storage/upload/13528"
TargetPath="\\testingserver2\Daily\Test\jcalvo\Mala_5039679\23310\hola.mxf"
TransferID="13528" xmlns="http://base.fims.tv"/>
    <storageType xmlns="http://base.fims.tv">online</storageType>
    <containerMimeType xmlns="http://base.fims.tv"/>
    <status xmlns="http://base.fims.tv">
      <ExtensionGroup/>
      <ExtensionAttributes/>
      <status>new</status>
    </status>
  </essencePlaceholder>
</addEssenceRequest>
```

The important parameters here is TransferID and StorageNodeID.

4.6.2 Output message, AddEssenceOperationAck

```
<addEssenceOperationAck      version="1_2_0"      xmlns="http://repository.fims.tv"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<timeStamp xmlns="">2019-03-06T16:13:31.9428653Z</timeStamp>
<operationID xmlns="">
  <resourceID
xmlns="http://base.fims.tv">9b0dd2ff-63bd-423e-bc4a-ce9e58543b76</resourceID>
  <location
xmlns="http://base.fims.tv">https://testingserver2:8080/wfservice/wip/9b0dd2ff-63b
d-423e-bc4a-ce9e58543b76</location>
  </operationID>
</addEssenceOperationAck>
```