



Back End Online Media Management and Distribution Platform

VP-Core / SURFmedia Core

Webservices Document

Version 1.5.3
April 2009

Back end online videodienst VP-Core

Webservices Interface Document

Behorende bij VP-Core versie 1.5.3

April 2009



Stichting Kennisnet Ict op School



Dit document is samengesteld door:

Harold Aling, Arjen Klop, Friso Postuma, Hugo Leisink, Robert Blanker, Joost Bloemsma Madcap BV/D2Groep

Madcap B.V.

Hakgriend 44a, 3371 KA Hardinxveld-Giessendam

Tel. (088 99 088 99)

Fax (088 99 088 98)

Internet: www.madcap.nl

Inhoudsopgave

1	Introductie.....	6
1.1	Doel.....	6
1.2	Leeswijzer.....	6
1.3	Afkortingen.....	6
1.4	Veranderingen aan rest calls [POST] VP-Core 1.1.6 en hoger.....	6
2	Inleiding.....	8
2.1	Beschrijving van de rest interfaces.....	10
3	Afspelen.....	12
3.1	Afspeel en/of download informatie van een video opvragen.....	12
3.2	Opvragen van een link naar een still.....	15
4	Identificatie van de Eindgebruikerapplicatie.....	16
5	Autorisatie.....	18
5.1	Opvragen van ACL regels van een Mediafile.....	18
5.2	Opslaan van ACL regels van een Mediafile.....	19
5.3	Verwijderen van ACL regels van een mediafile.....	22
5.4	Aanmaken van een autorisation group.....	23
5.5	Aanpassen van een autorisation group.....	24
5.6	Verwijderen van een autorisation group.....	24
5.7	Opvragen van een lijst van autorisation groups.....	25
5.8	Opvragen van een lijst van een enkele autorisation group.....	26
5.9	Aanmaken van een of meerdere hostnames binnen een autorisation group.....	27
5.10	Verwijderen van een of meerder hostnames in een autorisation group.....	28
5.11	Opvragen van alle hostnames in een autorisation group.....	30
5.12	Opvragen van ACL regels van een Collection.....	31
5.13	Opslaan van ACL regels van een Collection.....	32
5.14	Verwijderen van ACL regels van een collection.....	33
6	Upload.....	34
6.1	Upload via een eindgebruiker browser.....	34
6.2	Upload ticket aanvragen.....	35
6.3	Uploaden met upload_ticket (POST of PUT).....	36
6.4	Uploaden naar mediafile (PUT).....	37
6.5	FTP.....	38
6.6	Opvragen FTP gebruikers.....	38
6.7	Aanmaken van een FTP gebruiker.....	39
6.8	Opvragen van een FTP gebruiker.....	39
6.9	FTP gebruiker wijzigen.....	40
6.10	Verwijderen van een FTP gebruiker.....	40
7	Transcoding.....	41

7.1	<i>Transcoderen naar een default profiel.....</i>	41
7.2	<i>Transcoderen naar een gekozen profiel.....</i>	41
7.3	<i>Transcoderen met behulp van opgegeven parameters.....</i>	42
7.4	<i>Genereren van een still.....</i>	44
7.5	<i>Opvragen van jouw default (preview) transcode profiel.....</i>	45
8	<i>Jobs.....</i>	46
8.1	<i>Aanmaken van een transcodejob.....</i>	46
8.2	<i>Ophalen van de transcodeprofielen.....</i>	46
8.3	<i>Ophalen van een lijst van jobs voor een asset.....</i>	47
8.4	<i>Ophalen van een lijst van jobs voor een user.....</i>	47
8.5	<i>Opvragen van jobstatus.....</i>	49
8.6	<i>Verwijderen van een job.....</i>	49
9	<i>Assets.....</i>	51
9.1	<i>Lijst van assets ophalen.....</i>	51
9.2	<i>Details van een enkele asset opvragen.....</i>	56
9.3	<i>Lijst van collections waarin de asset voor komt opvragen.....</i>	57
9.4	<i>Asset aanmaken.....</i>	58
9.5	<i>Asset informatie aanpassen.....</i>	59
9.6	<i>Eigen metadata tag definitie toevoegen.....</i>	60
9.7	<i>Eigen metadata tag definitie verwijderen.....</i>	60
9.8	<i>Metatags opvragen.....</i>	61
9.9	<i>Metadata aan enkele asset toevoegen.....</i>	62
9.10	<i>Een enkele asset verwijderen.....</i>	64
9.11	<i>Meerdere assets tegelijkertijd verwijderen.....</i>	65
9.12	<i>Lijst van mediafiles behorende bij een enkele asset opvragen.....</i>	66
9.13	<i>De still behorende bij een enkele asset verwijderen.....</i>	66
9.14	<i>Het aantal assets tellen per applicatie, group en/of owner.....</i>	67
9.15	<i>Het aantal collections tellen waar de asset in voor komt.....</i>	67
9.16	<i>Supplement aanmaken.....</i>	68
9.17	<i>Supplementele data opslaan.....</i>	69
9.18	<i>Supplementele data van een enkele asset opvragen.....</i>	70
9.19	<i>Supplementele data verwijderen.....</i>	71
9.20	<i>Favoriet gebruikers asset / collection link toevoegen.....</i>	71
9.21	<i>Favoriet gebruikers asset / collection link verwijderen.....</i>	72
9.22	<i>Favorieten gebruiker opvragen.....</i>	73
9.23	<i>Aantal keren asset in favorieten.....</i>	73
9.24	<i>Aantal keren collection in favorieten.....</i>	73
10	<i>Collections.....</i>	75
10.1	<i>Lijst van collections opvragen.....</i>	75
10.2	<i>Details van een enkele collection opvragen.....</i>	77
10.3	<i>Lijst van assets behorende bij een enkele collection opvragen.....</i>	78
10.4	<i>Collection aanmaken.....</i>	82
10.5	<i>Collection informatie aanpassen.....</i>	83
10.6	<i>Een of meerdere assets aan een collectie koppelen.....</i>	85
10.7	<i>Een of meerdere asset/collectie koppelingen verwijderen.....</i>	85

10.8	Collectie verwijderen.....	86
10.9	Het aantal collections tellen per applicatie, group en/of owner.....	87
10.10	Voorbeeld request URL.....	87
11	Mediafiles.....	88
11.1	Mediafile aanmaken.....	88
11.2	Details van een enkele mediafile opvragen.....	89
11.3	Informatie van een enkele mediafile aanpassen.....	90
11.4	Een enkele mediafile verwijderen.....	91
11.5	Het aantal mediafiles tellen per applicatie, group en/of owner.....	91
12	Media Management overig	93
12.1	Overhevelen eigenaarschap.....	93
13	User / group management.....	94
13.1	Gebruikers opvragen.....	94
13.2	Groepen opvragen.....	95
13.3	Details van een groep opvragen.....	96
13.4	Details van een gebruiker opvragen.....	97
13.5	Gebruiker aanmaken.....	98
13.6	Groep aanmaken.....	98
13.7	Gebruiker verwijderen.....	99
13.8	Groep verwijderen	100
13.9	Gebruiker aanpassen.....	100
13.10	Groep aanpassen.....	101
14	Statistieken.....	102
14.1	Overzicht van de nieuwste mediafiles.....	102
14.2	Overzicht van de populairste mediafiles.....	103
14.3	Overzicht van de nieuwste collecties.....	104
14.4	Overzicht van de populairste collecties.....	105
14.5	Overzicht van de upload van mediafile per maand.....	105
14.6	Overzicht van schrijfruimte per container.....	106
14.7	Overzicht van de gevraagde streams.....	107
14.8	Overzicht van zoekacties.....	108
14.9	Overzicht van de populairste streams per periode.....	109
14.10	Overzicht van de meest gezochte woorden voor een periode.....	110
15	Overige rest calls.....	111
15.1	Opvragen lijst van error codes.....	111
16	CQL - Contextual Query Language.....	113
16.1	Basis van CQL.....	113
16.2	Context set.....	113
16.3	Relaties.....	113
16.4	Relation modifiers.....	115
16.5	Masking, wildcards.....	115
16.6	Booleans.....	116
16.7	Boolean modifiers.....	117

16.8 Asset zoek en sorteer indexen.....	117
16.9 Collection zoek en sorteer indexen (CQL alias vpx:collection).....	119
16.10 ACL regels zoeken (CQL alias vpx:acl).....	119
16.11 Zoek parameters die buiten CQL vallen.....	121
17 FAQ.....	122

1 Introductie

Dit document geeft een beschrijving van alle REST web services die VP-Core aan externe applicatie ontwikkelaars biedt.

1.1 Doel

Dit document geeft een gedetailleerde beschrijving van de REST interfaces en kan als basis dienen voor eindgebruikerapplicatie-ontwikkeling. De Doelgroep voor dit document zijn dan ook eindgebruikerapplicatie ontwikkelaars.

1.2 Leeswijzer

Een applicatie die gebruik wil maken van de diensten van VP-Core zal zich in eerste instantie moeten kunnen aanmelden (authenticatie), dus na het lezen van de inleiding is dit het eerste hoofdstuk waar aandacht aan besteed dient te worden. Na inloggen start een applicatie met een lege video-database, dus zal er een video geupload moeten worden. Het meest eenvoudig is het om een POST-upload formulier te implementeren. Na een geslaagde upload kan een afspeelverzoek (afspelen hoofdstuk) opgevraagd worden aan VP-Core om de video af te spelen, er zijn verschillende afspeelverzoeken mogelijk. Afhankelijk van de te ontwikkelen applicatie kunnen dan alle mediamanagement functies worden onderzocht, of bv transcoding.

Behorend bij VP-Core is een referentieapplicatie dat geschreven is in JAVA met het framework Liferay dat gebruikt kan worden als referentieapplicatie. Dit document is op een aantal plaatsen uitgebreid met voorbeeld PHP-code (bv bij authenticatie), dat gebruikt kan worden.

1.3 Afkortingen

Afkorting	Betekenis
API	Application Programmers Interface
CITE	Continuous Integration and Test Environment
IDE	Integrated Development Environment
OTAP	Ontwikkel, Test, Acceptatie, Productie
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
WSDL	Webservice Definition Language

1.4 Veranderingen aan rest calls [POST] VP-Core 1.1.6 en hoger

De parameters die voor rest calls met de methode POST zijn aangepast. Het is allemaal backwards compatible, maar documentatie is aangepast. Alle GET variabelen die voor 1.1.6 gebruikt werden met POST rest calls zijn allemaal omgezet naar POST. Dit betekent dat alle variabelen met de methode POST ook allemaal via de POST manier worden verstuurd. Vooral user_id was in de meeste gevallen een GET parameter, maar in sommige rest calls POST.

Dit is nu recht getrokken; alle POST rest calls hebben POST variabelen en alle GET rest calls hebben GET variabelen. Maar alle oude GET variabelen in de POST rest call blijven werken en zijn dus geheel backwards compatible.

2 Inleiding

VP-Core levert een schaalbaar en toekomst-bestendig platform voor het opslaan, onderhouden en serveren van beeldmateriaal. Binnen een gecontroleerde omgeving kunnen applicaties gebruik maken van een scala aan diensten, variërend van het eenvoudigweg opslaan, omschrijven, terugzoeken en afspelen van video's, tot en met het transcoderen van video-materiaal naar andere formaten en het aanbieden van video-collecties voor indexing door geavanceerde OAI-harvesters.

VP-Core is gebaseerd op Representational State Transfer (REST) architectuur: alle interacties met het systeem vinden plaats middels deze REST-calls. REST is een variant op het Simple Object Access Protocol (SOAP), waarbij verzoeken aan de server niet in XML maar in URL vorm gedaan worden. De antwoorden van de server zijn altijd in XML vorm.

Het XML antwoord is als volgt opgebouwd:

```
<?xml version="1.0"?>
<response>
  <header>
    <item_count>10</item_count>
    <item_count_total>23489</item_count_total>
    <item_offset/>
    <request_process_time>0.1064</request_process_time>
    <request_query_count>11</request_query_count>
    <request_result>success</request_result>
    <request_result_description></request_result_description>
    <request_result_id>601</request_result_id>
    <request_uri>[GET] asset?limit=10</request_uri>
    <vpx_version>1.0.1</vpx_version>
  </header>
  <items>
    <item id="1">
    ...
    </item>
    <item id="2">
    ...
    </item>
    ...
  </items>
</response>
```

Tag	Value	Description
response	n/a	container van het gehele XML antwoord
header	n/a	container van de request informatie
item_count	numeriek	aantal items in dit antwoord
item_count_total	numeriek	totaal aantal items wat aan het verzoek voldoet
item_offset	numeriek	de gebruikte offset

request_process_time	float	de tijd die VPX nodig heeft gehad om het antwoord te formuleren
request_result	alphanumeriek	'success' of 'error'
request_result_description	string	beschrijving behorende bij het resultaat
request_result_id	numeriek	het nummer van het resultaat.
request_uri	n/a	[http methode] en het verzoek richting vpx
vpx_version	versie	versienummer van het VPX platform
items	n/a	container die gevuld kan worden met de opgevraagde of aangemaakte items
item	n/a	dit veld bevat de opgevraagde of aangemaakte items

De volgende diensten worden door VP-Core aan de applicaties aangeboden.

Media beheer

- beheren van media (Create, Read, Update, Delete)
- beheren van Dublin Core metadata bij media
- opslag van applicatie-specifieke metadata
- uitlezen van technische metadata van media
- beveiliging van videomateriaal en metadata
- bijwerken van bestaande media met nieuwe media versies

Afspelen / streaming / download / live stream

- afspelen van media met een afspeelticket (bioscoopkaartje)
- controle op geldigheid tickets
- gestreamd afspelen van streamable media
- downloaden van media indien toegestaan
- afspeelverzoeken herleiden naar een live stream

Zoeken

- collecties voor indexering ter beschikking stellen aan OAI-PMH harvester
- media vinden middels zoekopdrachten

Autorisatie & Authenticatie

- challenge / response authenticatie op applicatie-niveau
- autorisatie op collectie, asset, media en metadata niveau

- autorisatie op webservice niveau

Transcoding

- transcoderen van media van en naar gangbare formaten: MPEG1/2, H.264, VC-1, WMV1/2, Flash video, Quicktime &c.
- mogelijkheid tot het gebruik van een standaard formaatkeuze
- mogelijkheid tot automatische transcoding na upload van media
- transcoding op aanvraag
- voortgangrapportage op transcoderingen
- asynchrone afhandeling van transcoderingen
- één bron na transcoding in meerdere formaten beschikbaar
- lineair opschaalbare transcoding farm beschikbaar

Upload

- opslaan van media middels W3C protocollen (HTTP POST/PUT)
- batch (ftp) operaties met XML gestuurde metadata

Diversen

- systeem statistieken
- systeem logging
- applicatie-, groep- en gebruiker quota
- genereren van stills uit media

2.1 Beschrijving van de rest interfaces

De rest van het document bestaat hoofdzakelijk uit een opsomming van de rest interface van VP-Core. Het bestaat uit een beschrijving, de rest definitie, een beschrijving van de parameters met hun betekenis, en een voorbeeld aanroep.

De Rest definitie bestaat uit een url aanroep waarin enkele parameters zijn opgenomen (tussen {}) en eindigd met de methode GET of POST. (bv. video/{asset_id}/play [GET]). De hostname is niet opgenomen in de beschrijving en is op moment van schrijven http://app.vpx.kennisnet.nl/

De paragraaf request parameters beschrijft in detail de parameters van de rest call, waarbij type 1 van GET of POST kan zijn. Met GET wordt bedoeld dat het in de query parameters dient te staan, met POST dat het via POST variabelen dient te worden meegegeven.

Bij sommige parameters is het mogelijk om “[]” toe te voegen waarmee meerdere parameters van dezelfde naam worden toegestaan. bv /asset?title=foo zoekt op alleen “foo” en de call /asset?title[]=foo&title[]=bar zoekt op assets met titel zowel “foo” als “bar”.

Bij de voorbeelden worden een aantal standaardwaarden gebruikt: `user_id=entree004`, `entree005`, `app_id=1` en `app_id=28`. Dit zijn volstrekt willekeurige voorbeelden en dienen slechts ter ondersteuning van het voorbeeld.

3 Afspelen

Het afspelen van een video is de basis van VP-Core, met behulp van een `asset_id` en een `mediafile_id` kan een video worden afgespeeld. VP-Core werkt momenteel met videostreaming servers die verschillende vormen van WMV, Flash en Mpeg kunnen afspelen.

3.1 Afspelen en/of download informatie van een video opvragen

Deze aanroep implementeert een afspelerzoek. Als belangrijkste parameter wordt een unieke mediafile id meegegeven. Als response wordt een URI naar de streaming video server teruggegeven (plain) al dan niet in een gewenst formaat (bv metafile of object). Het is ook mogelijk een download URL op te vragen die verwijst naar een download server. Een speciale afspeloctie is "still"; als er voor een asset een still is aangemaakt, dan kan met deze aanroep een link worden opgevraagd naar de still, voor een handiger verzoek van een still zie volgende restcall.

3.1.1 Request URL

/asset/{asset_id}/play [GET]

3.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	asset id	URL	verplicht
mediafile_id	alfanumeriek	mediafile id	GET	verplicht (niet verplicht indien profile_id wordt gebruikt)
profile_id	integer	Gebruik een mediafile van een bepaald profiel. 'mediafile_id' wordt genegeert indien profile_id is gezet.	GET	optioneel
user_id	string	user id van de persoon die de video wil afspelen, wordt gebruikt voor logging, en voor autorisatie.	GET	verplicht(*)
response	string	response type: <ul style="list-style-type: none"> ● plain (plain URL) ● metafile (metafile formaat, dit is afhankelijk van de type video) ● object (html object) ● download (download url) ● still (still) 	GET	optioneel, default=plain
group_id	string	group id van de persoon die de video wil afspelen, kan gebruikt worden voor autorisatie.	GET	optioneel

aut_domain	string	Het domein waarvan de aanvraag wordt gedaan, wordt gebruikt voor autorisatie, indien opgegeven. Het domein wordt gecontroleerd tegen de acl regels van de mediafiles, als domain voorkomt in de /acl/domain lijst, wordt de video afgespeeld. (zie hoofdstuk afscherming)	GET	optioneel
aut_realm	string	Gebruikt voor autorisatie, indien opgegeven (zie hoofdstuk afscherming)	GET	optioneel
start	integer	Start tijd van het video fragment in milliseconden. Moet kleiner zijn dan 24 uur (86.400.000)	GET	optioneel
autostart	boolean	De waarde van autostart in de object code aanpassen.	GET	optioneel (default = true)
duration	integer	Af te spelen lengte van het video fragment in milliseconden. Moet kleiner zijn dan 24 uur (86.400.000)	GET	optioneel
width	integer	De breedte van de video.	GET	optioneel, indien in combinatie met height niet gebruikt wordt, zal height door middel van ratio video worden berekent.
height	integer	De hoogte van de video.	GET	optioneel, indien in combinatie met width niet gebruikt wordt, zal width door middel van ratio video worden berekent.

(*) hoewel verplicht, wordt user_id niet gecontroleerd tegen de user lijst. De EGA is zelf verantwoordelijk voor de juiste vulling van dit veld.

3.1.3 Voorbeeld request URL

/asset/212swfkk7pzWZGjtTRM/?mediafile_id=TYBaMHCP7bLKG8BdlcbYQ&user_id=entree004

3.1.4 Voorbeeld output

```
<items>
  <item id="1">
    <output>http://wms.acceptatie.streaming.kennisnet.nl/vpx-test/EITiElxvvLXfhYLnnp1B0nsCJEj</output>
    <content_type></content_type>
  </item>
</items>
```

3.1.5 Response fields

Field	Description
-------	-------------

value	URL naar de opgevraagde content
output	afhankelijk van het opgegeven response.
content_type	bij response = metafile.

3.2 Opvragen van een link naar een still

Deze aanroep implementeert een stillverzoek. Een asset_id wordt meegegeven als parameter. Het resultaat is een url naar een image op een webserver van VP-Core. Deze url kan eenvoudig gebruikt worden in bv een tag van de applicatie.

3.2.1 Request URL

/asset/{asset_id}/still [GET]

3.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	asset id	URL	verplicht
user_id	string	user id, wordt gebruikt voor logging.	GET	verplicht

3.2.3 Voorbeeld request URL

/asset/6c704fbd4ef58f2447fd1a3e7/still?user_id=entree004

3.2.4 Voorbeeld response

```
<items>
  <item id="1">
    <output>http://download.vpx.kennisnet.nl/still/MhS3b2rOUdsoYDoUFEBxT8</output>
    <content_type></content_type>
  </item>
</items>
```

3.2.5 Response fields

Field	Description
value	Url naar de image op de download server

4 Identificatie van de Eindgebruikerapplicatie

Alvorens een Eindgebruiker Applicatie (EGA) gebruik kan maken van VP-Core, dient deze zichzelf eerst middels REST calls te identificeren. Bij de eerste request van het identificatieproces wordt een cookie teruggegeven door de VP-Core server (via de 'Set-Cookie' HTTP header string). Deze cookie bevat een sessie-id, welke bij iedere volgende request ook weer als cookie dient terug te worden gegeven (via de 'Cookie' HTTP header string). De EGA krijgt namelijk binnen de bijbehorende sessie toegang tot VP-Core.

Identificatie van de Eindgebruiker Applicatie (EGA) gebeurt middels het DBUS protocol. Dit protocol is eigenlijk een tekstprotocol, terwijl VP-Core gebruikt maakt van HTTP. Om toch gebruik te kunnen maken van DBUS is de volgende wijziging gemaakt: de client stuurt de DBUS berichten als HTTP POST data (variabele 'dbus') en de server stuurt de antwoorden in XML terug (tag 'dbus'). De URL voor de POST requests is `http://<vpx-url>/login`.

Een beschrijving van het DBUS protocol is te vinden op de volgende webpagina:

`http://dbus.freedesktop.org/doc/dbus-specification.html`

VP-Core gebruikt de challenge-response methode DBUS_COOKIE_SHA1 als identificatie mechanisme binnen DBUS.

1. De EGA stuurt "AUTH DBUS_COOKIE_SHA1 <gebruikersnaam>" naar VP-Core, waarna deze de volgende informatie terug stuurt: "DATA vpx 0 <challenge>"
 - C: AUTH DBUS_COOKIE_SHA1 username
 - S: DATA vpx 0 b8225c6d20a6bef3d8ccfb3b5a30062
2. VP-Core heeft samen met bovenstaande <challenge> ook een cookie gestuurd in de vorm van: "SESS421aa90e079fa326b6494f812ad13e79=e490f6b992e33a41b9bbb7f101dbd300; expires=Thu, 08 May 2008 16:03:38 GMT; path=/" . Bij alle requests naar VP-Core moet vanaf nu het sessie-id "SESS421aa90e079fa326b6494f812ad13e79=e490f6b992e33a41b9bbb7f101dbd300" in de HTTP header meegegeven worden als cookie.
3. De EGA verzint een willekeurige tekst <random> en berekent de response string: `sha1(<challenge>:<random>:<wachtwoord>)` . Het resultaat hiervan wordt in de volgende vorm teruggestuurd naar VP-Core: "DATA <random> <response>". Indien de response wordt geaccepteerd, geeft VP-Core "OK vpx" als antwoord, anders "REJECTED".
 - C: DATA 4b9d547c49c 58a345b427acde144b9d547c49cd1a2ec4a26190
 - S: OK vpx

Een voorbeeld in PHP van hoe een client zich kan authenticeren is opgenomen in appendix A.

4.1.1 Request url

`/login [POST]`

4.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
dbus	string	dbus authenticatie informatie zoals uitgelegd in	POST	verplicht

		bovenstaande beschrijving		
--	--	---------------------------	--	--

4.1.3 Voorbeeld request URL

/login

formulier data: dbus="AUTH DBUS_COOKIE_SHA1 <gebruikersnaam>"

4.1.4 Response fields

Field	Description
dbus	dbus authenticatie informatie

5 Autorisatie

VP-Core biedt de mogelijkheid om videobestanden af te schermen voor andere gebruikers. De toegang wordt gezet met de volgende ACL (access control lists) regels. De rechten worden op mediafile niveau gezet, waarmee de mogelijkheid wordt geboden om bv een aantal lage kwaliteitsvideo's onder een asset publiek te maken en een hoge kwaliteitsversie af te schermen voor een groep gebruikers. De controle vindt plaats bij de afspeel rest call (playproxy). Als een mediabestand wordt geupload, is het niet afschermd, oftewel voor iedereen af te spelen. Op het moment dat er 1 regel wordt gezet, verandert deze status naar volledig afgeschermd afgezien van de opgegeven afschermregels.

De controle op afspelen kan worden gezet op:

- domein;
Een internet domein naam, bv geef alleen “surfnet.nl” toegang. Alle subdomains krijgen ook toegang tot de opgegeven domeinen, bv flex.surfnet.nl heeft in dit voorbeeld ook toegang. Het is mogelijk om een groep domeinnamen te maken, bv: ACADEMIA.group, waarin meerdere domeinen zijn opgenomen. Een domain-group dient te eindigen met '.group'.
- realm;
bv jan@xs4all.nl, of @surfnet.nl. In het eerste geval moet de realm exact matchen aan de aanvrager van de video. Het tweede geval is flexibeler: alle gebruikers die voldoen aan *@surfnet.nl alsmede de eventuele subdomains krijgen toegang (bv piet@surfnet.nl, jan@flex.surfnet.nl, joris@flex1.z33.surfnet.nl). Het is mogelijk om een groep van realms te maken, bv: @ACADEMIA.group, waarin meerdere realms zijn opgenomen. Een realm-group dient te starten met een '@' en te eindigen met '.group'.
- users;
Dit zijn VP-Core users, en wordt gecontroleerd tegen het “user_id” veld bij de afspeel opdracht (video/./play)
- groepen;
Dit zijn VP-Core users en wordt gecontroleerd tegen het “group_id” veld bij de afspeel opdracht (video/./play)
- applicaties;
Dit is een speciale variant van de afspeelrechten. Het is mogelijk om andere applicaties die bij VP-Core zijn aangesloten afspeel-toegang te geven tot de video's. Hiertoe moet het applicatie id bekend zijn. Binnen de andere applicatie is de video dan publiek te bekijken, mits er geen realm/domain rechten zijn gezet. Als deze wel gezet zijn op de mediafile, moet de andere EGA ook voldoen aan de domain/realm rechten om de mediafile te kunnen bekijken. User/group rechten worden voor andere EGA's genegeert. Dit omdat users/groepen per EGA uniek zijn; een mediafile kan dus nooit een user/group van een andere EGA als owner/recht hebben.

5.1 Opvragen van ACL regels van een Mediafile

Een mediafile is een fysiek (video) bestand. Een asset kan meerdere mediafiles bevatten. Deze rest call genereert een xml-output van alle acl-regels die op de mediafile gelden.

5.1.1 Request url

```
/mediafile/{mediafile_id}/acl [GET]
```

5.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	mediafile id, unieke identifier van de mediafile.	URL	verplicht
user_id	string	user id, wordt bepaald om te controleren of de aanvrager de rechten heeft om autorisatie op te vragen. Alleen de eigenaar van de mediafile heeft recht op deze mogelijkheid.	GET	verplicht

5.1.3 Voorbeeld request URL

/mediafile/6c704fbd4ef58f2447fd1a3e7/acl?user_id=entree004

5.1.4 Voorbeeld response

```
<items>
  <item id="1">
    <aut_realm>@teleblik.nl</aut_realm>
  </item>
  <item id="2">
    <aut_domain>ACADEMIA.group</aut_domain>
  </item>
</items>
```

5.1.5 Response fields

Field	Description
items	xml output van acl regels.

5.2 Opslaan van ACL regels van een Mediafile

Met deze call worden de acl regels op een mediafile gewijzigd, alle bestaande acl-regels op deze mediafile worden vervangen door de meegestuurde regels. Met behulp van 'replace' setting op FALSE te zetten worden de bestaande regels niet vervangen.

5.2.1 Request URL

/mediafile/{mediafile_id}/acl [POST]

5.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	mediafile id	URL	verplicht
user_id	string	user id, moet gelijk zijn aan de eigenaar van	POST	verplicht

		mediafile.		
replace	bool	'TRUE' of 'FALSE' indien de huidige rechten moeten worden vervangen. Standaard is TRUE. Definieer als 'FALSE' indien je de oude rechten wilt behouden.	POST	optioneel
aut_app[]	string	application(s) die toegang krijgen tot de mediafile	POST	optioneel (POST)
aut_user[]	string	user(s) die toegang krijgen tot de mediafile	POST	optioneel (POST)
aut_group[]	string	groep(en) die toegang krijgen tot de mediafile	POST	optioneel (POST)
aut_domain[]	string	domein(en) die toegang krijgen tot de mediafile	POST	optioneel (POST)
aut_realm[]	string	realm(s) die toegang krijgen tot de mediafile	POST	optioneel (POST)

5.2.3 Voorbeeld request URL

Voorbeeld 1:

/mediafile/6c704fbd4ef58f2447fd1a3e7/acl

POST:

user_id=entree004
aut_user[]=entree005

Hiermee geeft user “entree004” leesrechten aan user “entree005”. Bij het aanroepen van de /asset/play restcall wordt de autorisatie-controle uitgevoerd. De volgende playproxy aanroepen doen dan:

```
video/23d4e45f523823df2/play? mediafile_id=6c704fbd4ef58f2447fd1a3e7&response=plain&
user_id=entree004 : mag afspelen
user_id=entree005 : mag afspelen
user_id=entree006 : mag niet afspelen
```

Voorbeeld 2:

/mediafile/6c704fbd4ef58f2447fd1a3e7/acl

POST:

user_id=entree004
aut_domain[]=surfnet.nl

Hiermee geeft user “entree004” leesrechten aan iedereen die van het domein surfnet.nl afkomen. bv:

```
video/23d4e45f523823df2/play? mediafile_id=6c704fbd4ef58f2447fd1a3e7&response=plain&
user_id=entree004 : mag afspelen
user_id=entree005 : mag niet afspelen
domain=xs4all.nl : mag niet afspelen
```

```

group_id=root : mag niet afspelen
domain=surfnet.nl : mag wel afspelen
domain=flex.surfnet.nl : mag wel afspelen
domain=.nl : mag niet afspelen
domain=xs4all.nl&user_id=entree004 : mag wel afspelen
domain=surfnet.nl&user_id=entree005 : mag wel afspelen

```

Voorbeeld 3:

```

/mediafile/6c704fbd4ef58f2447fd1a3e7/acl
POST:
user_id=entree004
aut_domain[]=surfnet.nl
aut_domain=xs4all.nl
aut_realm=@vu.nl

```

Hiermee geeft user “entree004” leesrechten aan iedereen die van het domein surfnet.nl en xs4all.nl afkomen en de personen die passen in de realm @vu.nl bv:

```

video/23d4e45f523823df2/play? mediafile_id=6c704fbd4ef58f2447fd1a3e7&response=plain&
aut_domain=xs4all.nl : mag afspelen
group_id=surf : mag niet afspelen
domain=surfnet.nl : mag wel afspelen
domain=flex.surfnet.nl : mag wel afspelen
domain=.nl : mag niet afspelen
realm=jan@xs4all.nl : mag niet afspelen
realm=piet@vu.nl : mag wel afspelen
domain=surfnet.nl&user_id=entree005 : mag wel afspelen

```

Voorbeeld 4:

```

/mediafile/6c704fbd4ef58f2447fd1a3e7/acl
POST:
user_id=entree004
aut_app[]=2
aut_domain=xs4all.nl

```

Hiermee geeft user “entree004” leesrechten aan iedereen die van het domein xs4all.nl afkomt binnen de eigen applicatie. Daarnaast krijgen alle gebruikers van applicatie 2 ook toegang.

```

video/23d4e45f523823df2/play? mediafile_id=6c704fbd4ef58f2447fd1a3e7&response=plain&
domain=xs4all.nl : mag afspelen
group_id=surf : mag niet afspelen, indien eigen applicatie
group_id=surf : mag niet afspelen, indien applicatie 2

```

5.2.4 Voorbeeld response

```

<response>
  <header>
    <item_count>1</item_count>
    <request_process_time>0.0649</request_process_time>
    <request_query_count>17</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST]mediafile/T6WPMgzmsT6njMQepTNq8AgUxJG/acl?user_id=1</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items>
    <item id="1">
      <aut_domain>

```

```

                <value>surfnet.nl</value>
                <result>success</result>
                <result_id>601</result_id>
                <result_description/>
            </aut_domain>
        </item>
    </items>
</response>

```

5.2.5 Response fields

Field	Description
value	true/false

5.3 Verwijderen van ACL regels van een mediafile

Verwijdert alle afschermregels op een mediafile en maakt het bestand publiekelijk afspeelbaar.

5.3.1 Request url

/mediafile/{mediafile_id}/acl/delete [POST]

5.3.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	mediafile id, unieke identifier van de mediafile.	URL	verplicht
user_id	string	user id, wordt bepaald om te controleren of de aanvrager de rechten heeft om autorisatie te verwijderen.	POST	verplicht

5.3.3 Voorbeeld request URL

/mediafile/6c704fbd4ef58f2447fd1a3e7/acl/delete?user_id=entree004

5.3.4 Voorbeeld response

```

<response>
  <header>
    <item_count>0</item_count>
    <request_process_time>0.0065</request_process_time>
    <request_query_count>13</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST] mediafile/T6WPMgzmsT6njMQepTNq8AgUxJG/acl/delete?user_id=1</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
</items/>
</response>

```

5.3.5 Response fields

Field	Description
value	true/false

5.4 Aanmaken van een autorisation group

Met deze call kun je een autorisation group aanmaken. Een group domeinen of realms dienen met '.group' te eindigen, om zich te kunnen onderscheiden van domeinen en realms. Een Realm groep dient te starten met een '@'.

5.4.1 Request url

/autorisation_group/create [POST]

5.4.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
groupname	string	de naam van de group die aangemaakt moet worden. Moet eindigen met '.group'. Een group van het type 'realm' dient te starten met een '@'.	POST	verplicht
group_type	'domain' of 'realm'	het type group dat aangemaakt moet worden.	POST	verplicht

5.4.3 Voorbeeld request URL

```
/autorisation_group/create
POST
groupname=@academia.group
group_type=realm
```

5.4.4 Voorbeeld response

```
<response>
  <header>
    <item_count>0</item_count>
    <request_process_time>0.0065</request_process_time>
    <request_query_count>13</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST] mediafile/T6WPMgzmsT6njMQepTNq8AgUxJG/acl/delete?user_id=1</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
</items/>
</response>
```

5.4.5 Response fields

Field	Description
-------	-------------

n/a	Bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie
-----	---

5.5 Aanpassen van een autorisation group

Met deze call kun je een bestaande autorisation group aanpassen.

5.5.1 Request url

/autorisation_group/{groupname} [POST]

5.5.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
groupname	string	de naam van de group die aangepast moet worden.	URL	verplicht
group_type	'domain' of 'realm'	het type group van de group	POST	optioneel
new_groupname	string	de nieuwe naam van de group	POST	optioneel

5.5.3 Voorbeeld request URL

/autorisation_group/academia.group
 POST:
 new_groupname=ho.group
 group_type=domain

5.5.4 Voorbeeld response

```
<response>
  <header>
    <item_count>0</item_count>
    <request_process_time>0.0056</request_process_time>
    <request_query_count>11</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST] autorisation_group/Group_A</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items/>
</response>
```

5.5.5 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

5.6 Verwijderen van een autorisation group

Met deze call kun je een autorisation group verwijderen.

5.6.1 Request url

/autorisation_group/{groupname}/delete [POST]

5.6.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
groupname	string	de naam van de group die aangemaakt moet worden.	URL	verplicht
delete	string	Met de waarde 'cascade' wordt alle koppelingen verwijderd met deze groep. Indien 'delete' niet deze waarde heeft, kan alleen de groep worden verwijderd indien deze geen koppelingen meer heeft.	POST	optioneel

5.6.3 Voorbeeld request URL

/autorisation_group/ho.group/delete

Formulier data;

delete=cascade

5.6.4 Voorbeeld response

```
<response>
  <header>
    <item_count>0</item_count>
    <request_process_time>0.0347</request_process_time>
    <request_query_count>19</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST] autorisation_group/Group_A/delete</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items/>
</response>
```

5.6.5 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

5.7 Opvragen van een lijst van autorisation groups

Met deze call kun je een lijst van autorisation groups opvragen.

5.7.1 Request url

/authorisation_group [GET]

5.7.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	het maximum aantal groups dat moet worden weergegeven (max 200)	GET	verplicht
offset	numeriek	geef <limit> items terug vanaf <offset>	GET	optioneel

5.7.3 Voorbeeld request URL

/authorisation_group?limit=200&offset=400

5.7.4 Voorbeeld response

```
<response>
  <header>
    <item_count>2</item_count>
    <item_count_total>2</item_count_total>
    <request_process_time>0.0306</request_process_time>
    <request_query_count>10</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[GET] authorisation_group/?limit=10</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items>
    <item id="1">
      <aut_group_name>HO.group</aut_group_name>
      <aut_group_type>DOMAIN</aut_group_type>
    </item>
    <item id="2">
      <aut_group_name>@academia.group</aut_group_name>
      <aut_group_type>REALM</aut_group_type>
    </item>
  </items>
</response>
```

5.7.5 Response fields

Field	Description
aut_group_name	naam van de autorisation group
aut_group_type	type van de autorisation group

5.8 Opvragen van een lijst van een enkele autorisation group

Met deze call kun je de details van een enkele autorisation group opvragen.

5.8.1 Request url

/authorisation_group/{groupname} [GET]

5.8.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
groupname	string	de naam van de group die weergegeven moet worden.	URL	verplicht

5.8.3 Voorbeeld request URL

/authorisation_group/@ACADEMIA.group [GET]

5.8.4 Voorbeeld response

```
<response>
  <header>
    <item_count>1</item_count>
    <request_process_time>0.006</request_process_time>
    <request_query_count>12</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[GET] authorisation_group/@ACADEMIA.group</request_uri>
    <vpx_version>1.0.3</vpx_version>
  </header>
  <items>
    <item id="1">
      <groupname>@ACADEMIA.group</groupname>
      <group_type>REALM</group_type>
    </item>
  </items>
</response>
```

5.8.5 Response fields

Field	Description
groupname	naam van de autorisation group
group_type	type van de autorisation group

5.9 Aanmaken van een of meerdere hostnames binnen een autorisation group

Met deze call kun je een of meerdere hostnames aanmaken binnen een autorisation group.

5.9.1 Request url

/authorisation_group/{groupname}/hostname/create [POST]

5.9.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
-----------	--------	--------------	---------	---------------------

groupname	string	de naam van de groep waaraan de hostname gekoppeld moet worden.	URL	verplicht
hostname of hostname[]	string	de hostname die aan de groep gekoppeld moet worden. Middels de '['] notatie kunnen meerdere hostnames tegelijkertijd aangemaakt worden.	POST	verplicht

5.9.3 Voorbeeld request URL

/authorisation_group/ho.group/hostname/create

POST:

hostname[]=kennisnet.nl

hostname[]=surfnet.nl

5.9.4 Voorbeeld response

```

<response>
  <header>
    <item_count>2</item_count>
    <request_process_time>0.0073</request_process_time>
    <request_query_count>14</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST] authorisation_group/ho.group/hostname/create</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items>
    <item id="1">
      <hostname>kennisnet.nl</hostname>
      <result>success</result>
      <result_id>601</result_id>
      <result_description/>
    </item>
    <item id="2">
      <hostname>surfnet.nl</hostname>
      <result>success</result>
      <result_id>601</result_id>
      <result_description/>
    </item>
  </items>
</response>

```

5.9.5 Response fields

Field	Description
hostname	hostname die aangemaakt wordt
result	Het resultaat van de actie
result_id	Het resultaat ID van de actie

result_description	De beschrijving van het resultaat van de actie
--------------------	--

5.10 Verwijderen van een of meerder hostnames in een autorisation group

Met deze call kun je een of meerdere hostnames in een autorisation group verwijderen.

5.10.1 Request url

/autorisation_group/{groupname}/hostname/delete [POST]

5.10.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
groupname	string	de naam van de group die aangemaakt moet worden.	URL	verplicht
hostname of hostname[]	string	de hostname die aan de group gekoppeld moet worden middels de '[]' notatie kunnen meerder hostnames tegelijkertijd aangemaakt worden	POST	verplicht

5.10.3 Voorbeeld request URL

/autorisation_group/ho.group/hostname/delete

POST:

hostname[]=Host_1

hostname[]=Host_2

5.10.4 Voorbeeld response

```
<response>
  <header>
    <item_count>2</item_count>
    <request_process_time>0.0086</request_process_time>
    <request_query_count>16</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST] autorisation_group/ho.group/hostname/delete</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items>
    <item id="1">
      <hostname>Host_1</hostname>
      <result>success</result>
      <result_id>601</result_id>
      <result_description/>
    </item>
    <item id="2">
      <hostname>Host_2</hostname>
      <result>success</result>
      <result_id>601</result_id>
      <result_description/>
    </item>
  </items>
</response>
```

```
</items>
</response>
```

5.10.5 Response fields

Field	Description
hostname	hostname die verwijderd wordt
result	Het resultaat van de actie
result_id	Het resultaat ID van de actie
result_description	De beschrijving van het resultaat van de actie

5.11 Opvragen van alle hostnames in een autorisation group

Met deze call kun je alle hostnames binnen een autorisation group opvragen.

5.11.1 Request url

```
/autorisation_group/{groupname}/hostname [GET]
```

5.11.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	het maximum aantal hostnames dat moet worden weergegeven (max 200)	GET	verplicht
offset	numeriek	geef <limit> items terug vanaf <offset>	GET	optioneel

5.11.3 Voorbeeld request URL

```
/autorisation_group/academia.grou/hostname?limit=200
```

5.11.4 Voorbeeld response

```
<response>
  <header>
    <item_count>2</item_count>
    <item_count_total>2</item_count_total>
    <request_process_time>0.0048</request_process_time>
    <request_query_count>11</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[GET] autorisation_group/academia.group/hostname/?limit=10</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items>
    <item id="1">
      <hostname>Host_1</hostname>
    </item>
    <item id="2">
```

```

                <hostname>Host_2</hostname>
            </item>
        </items>
    </response>

```

5.11.5 Response fields

Field	Description
hostname	hostname

5.12 Opvragen van ACL regels van een Collection

Een collection is een verzameling assets. Deze rest call genereert een xml-output van alle acl-regels die op de collection gelden.

5.12.1 Request url

```
/collection/{coll_id}/acl [GET]
```

5.12.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
coll_id	alfanumeriek	collection id, unieke identifier van de collection.	URL	verplicht
user_id	string	user id, wordt bepaald om te controleren of de aanvrager de rechten heeft om autorisatie op te vragen. Alleen de eigenaar van de collection heeft recht op deze mogelijkheid.	GET	verplicht

5.12.3 Voorbeeld request URL

```
/collection/58/acl?user_id=entree004
```

5.12.4 Voorbeeld response

```

<response>
  <header>
    <item_count>1</item_count>
    <request_process_time>0.006</request_process_time>
    <request_query_count>12</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[GET] collection/425/acl?user_id=brin_9274712</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items>
    <item id="1">
      <aut_app>3</aut_app>
    </item>
  </items>
</response>

```

5.13 Opslaan van ACL regels van een Collection

Met deze call worden de acl regels op een collection gewijzigd, alle bestaande acl-regels op deze collection worden vervangen door de meegestuurde regels. Met behulp van 'replace' setting op FALSE te zetten worden de bestaande regels niet vervangen.

5.13.1 Request URL

/collection/{coll_id}/acl [POST]

5.13.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
coll_id	alfanumeriek	coll id	URL	verplicht
user_id	string	user id, moet gelijk zijn aan de eigenaar van collection.	POST	verplicht
replace	bool	'TRUE' of 'FALSE' indien de huidige rechten moeten worden vervangen. Standaard is TRUE. Defineer als 'FALSE' indien je de oude rechten wilt behouden.	POST	optioneel
aut_app[]	string	application(s) die toegang krijgen tot de collection.	POST	optioneel

5.13.3 Voorbeeld request URL

Voorbeeld 1:

/collection/58/acl

POST:

user_id=entree004

aut_app[]=2

5.13.4 Voorbeeld response

```
<response>
  <header>
    <item_count>1</item_count>
    <request_process_time>0.0076</request_process_time>
    <request_query_count>14</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST] collection/58/acl?user_id=entree004</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
  <items>
    <item id="1">
      <aut_app>
        <value>2</value>
        <result>success</result>
        <result_id>601</result_id>
        <result_description/>
      </aut_app>
    </item>
  </items>
</response>
```

```

        </item>
      </items>
    </response>

```

5.14 Verwijderen van ACL regels van een collection

Verwijderd alle afscherm-/toegangsregels op een collection.

5.14.1 Request url

/collection/{collection_id}/acl/delete [POST]

5.14.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
coll_id	alfanumeriek	collection id, unieke identifier van de collection.	URL	verplicht
user_id	string	user id, wordt bepaald om te controleren of de aanvrager de rechten heeft om autorisatie te verwijderen.	POST	verplicht

5.14.3 Voorbeeld request URL

/collection/58/acl/delete?user_id=entree004

5.14.4 Voorbeeld response

```

<response>
  <header>
    <item_count>0</item_count>
    <request_process_time>0.0063</request_process_time>
    <request_query_count>12</request_query_count>
    <request_result>success</request_result>
    <request_result_description/>
    <request_result_id>601</request_result_id>
    <request_uri>[POST] collection/425/acl/delete?user_id=entree004</request_uri>
    <vpx_version>1.1-rc2</vpx_version>
  </header>
</items/>
</response>

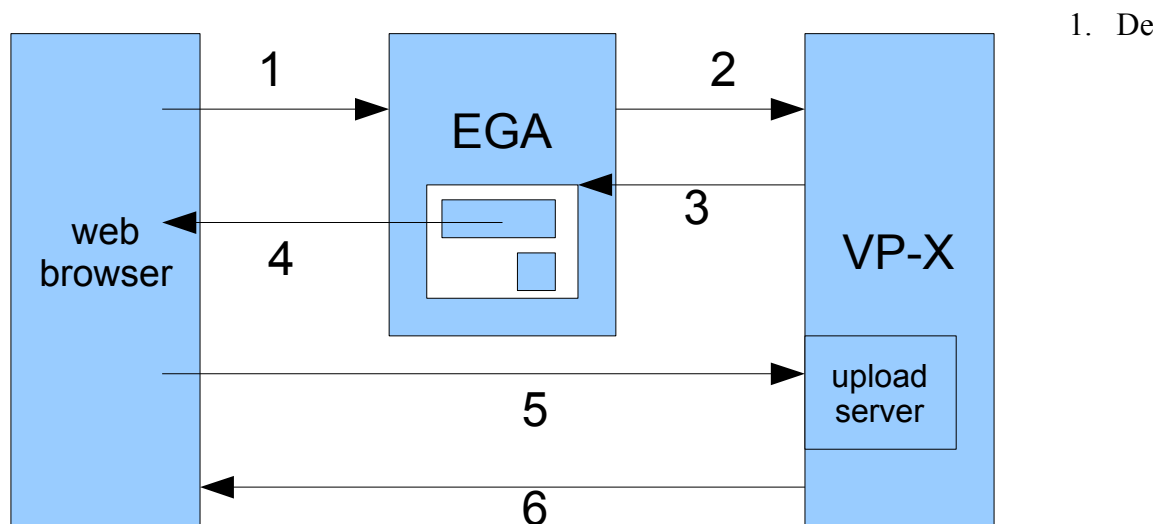
```

6 Upload

VP-Core biedt 3 methoden om videobestanden te uploaden; via de eindgebruikerapplicatie, rechtstreeks via de eindgebruiker-browser middels een POST upload formulier of FTP. De methode via de eindgebruikerapplicatie gaat m.b.v. HTTP PUT, en heeft geen restricties voor de grootte van de bestanden. Het POST formulier heeft een restrictie van 100 Mb. De methode om een eindgebruiker rechtstreeks middels POST te laten uploaden gaat in 2 stappen, eerst een ticket aanvragen en vervolgens een bestand laten uploaden.

6.1 Upload via een eindgebruiker browser

Het volgende diagram geeft stap voor stap de procedure weer van uploaden van een video naar VPX.



gebruiker klikt op een link om een video te uploaden

2. De EGA vraagt een ticket aan bij VP-Core. Dit is niets meer dan een string waarmee VP-Core de eindgebruiker kan identificeren. Het ticket wordt gekoppeld aan een asset/mediafile zodat VP-Core weet wat met het geüploade bestand moet gebeuren. Het ticket is 1 uur geldig en wordt ongeldig zodra er 1 upload mee is gedaan.
3. VP-Core geeft het ticket terug.
4. De EGA presenteert een html formulier met een action URL naar de VP-Core upload server. De URL van de upload-server komt als informatie bij het aanvragen van de ticket terug. In het formulier moet de EGA ook een terugkeer opgeven (zie stap 6).
5. De eindgebruiker vult in het formulier zijn bestand in, en klikt op submit. Vervolgens start de upload rechtstreeks naar de upload-server.
6. Als VP-Core het bestand heeft ontvangen, stuurt het een redirect naar de eindgebruiker om weer op de EGA website terug te komen.

6.2 Upload ticket aanvragen

Om via POST een bestand op te kunnen slaan moet er eerst een upload ticket op worden gevraagd. Voor het uploaden van een nieuw bestand, moet eerst een nieuw asset_id en mediafile_id aangemaakt worden (zie Media-management hoofdstuk). Het ticket is 60 minuten geldig en is alleen te gebruiken voor de mediafile waar hij voor aangevraagd is.

6.2.1 Request URL

/mediafile/{mediafile_id}/uploadticket/create [GET]

6.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	id van een bestaande originele (dus geen transcode) of lege mediafile	URL	verplicht
user_id	string	user id, wordt bepaald om te controleren of de aanvrager de rechten en quota heeft om de mediafile te uploaden	GET	verplicht
group_id	string	group id, wordt gebruikt voor een eventuele quota check.	GET	optioneel

6.2.3 Voorbeeld request URL

/mediafile/6c704fbd4ef58f2447fd1a3e7/upload_ticket?user_id=t.gebruiker

6.2.4 Response fields

Field	Description
upload_ticket	het door VPX gegenereerde alfanumerieke ticket
action	De action die opgeven moet worden in het html-formulier, hierin is tevens de juiste upload-server opgenomen. Zie voorbeeld in het volgende paragraaf.
uploadprogress_url	Deze url wordt gebruikt om tijdens de upload de status van een upload op te vragen. Alleen via POST uploads kan deze URL gebruikt worden. De informatie die terug komt is een JSON formaat (application/json) en kan via JQuery worden verwerkt via de client browser.

6.2.5 Voorbeeld response

```
<items>
  <item id="1">
    <upload_ticket>j89y492b84</upload_ticket>
    <action>http://upload.vpx.kennisnet.nl/mediafile/upload?upload\_ticket=j89y492b84</action>
    <uploadprogress_url>http://upload.vpx.kennisnet.nl/mediafile/uploadprogress</uploadprogress_url>
  </item>
</items>
```

6.3 Uploaden met *upload_ticket* (POST of PUT)

Middels deze POST of PUT request kan een mediafile naar VP-Core gestuurd worden. Hiervoor is alleen een geldige upload ticket nodig. Eventueel kan de eindgebruiker worden doorgestuurd naar een andere URL nadat de upload voltooid of afgebroken is. Deze upload moet naar de server gestuurd worden dat is meegegeven als informatie bij het aanvragen van de ticket.

Deze request kan via POST en PUT aangesproken worden. De parameters verschillen wel minimaal per methode. Zo is bij PUT een bestandsnaam verplicht en bij POST het 'file' formulier veld.

6.3.1 Request URL

/mediafile/upload [POST] of [PUT]

6.3.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
upload_ticket	alfanumeriek	upload ticket, welke verkregen kan worden door via de EGA een mediafile aan te maken.	POST	verplicht
transcode[]	numeriek	een (of meerdere) transcode profielen waar de mediafile naartoe getranscodeerd moet worden na de upload.	POST	optioneel
transcode_inherits_acl	bool	Waarde 'true'; Bij toekomstige transcodes worden alle mediafile acl rechten gecopieerd op de nieuwe transcoded mediafile.	POST	optioneel; default = 'false'
redirect_uri	uri	een URI waarnaar de eindgebruiker wordt doorgestuurd na upload.	POST	optioneel
file	file	HTML FORM field van het type 'file'.	POST	verplicht indien POST
retranscode	boolean	indien 'retranscode' niet gezet is, zullen alle bestaande transcodes van de originele mediafile verloren gaan. Als 'retranscode' 'true' is, dan zullen er nieuwe transcode jobs aangemaakt worden, die de bestaande transcodes vervangen.	POST	optioneel; default = 'false'
filename	string	De bestandsnaam die de mediafile weer aan neemt als hij gedownload wordt.	POST	verplicht indien PUT
create_still	boolean	Specificeer 'true' wanneer je een still gegenereerd wilt hebben na de upload en analyse.	POST	optioneel; default = 'false'

completed_url	url	<p>Door deze URL te definiëren geeft de EGA bouwer de mogelijkheid om op deze URL een status rapport te krijgen van de upload. De parameters die mee gegeven worden zijn; upload_ticket status_code</p> <p>De upload_ticket wordt gebruikt om de upload te indentificeren. De status_code is een VPX code, zie appendix B voor de lijst van codes.</p> <p>Het type is URL en niet URI is omdat deze parameter alleen http:// of https:// als schema mag zijn.</p>	POST	optioneel
---------------	-----	---	------	-----------

6.3.3 Voorbeeld request URL

http://upload.vpx.kennisnet.nl/mediafile/upload
 POST:
 upload_ticket=j89y492b84
 transcode[]=1
 transcode[]=4

6.3.4 Voorbeeld html-formulier

```
<form method="post" enctype="multipart/form-data" action="http://upload.vpx.kennisnet.nl/mediafile/upload" />
<input type="file" name="file" />
<input type="hidden" name="upload_ticket" value="j89y492b84" />
<input type="hidden" name="redirect_uri" value="http://www.example.com" />
<input type="submit" value="Upload" />
</form>
```

6.3.5 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven, maar er wordt een browser redirect gestuurd naar de terugkeer URI. Indien geen terugkeer URI is opgegeven worden een xml response getoond met naast de standaard header de request informatie.

6.4 Uploaden naar mediafile (PUT)

Deze aanroep implementeert de daadwerkelijke upload naar een mediafile. De upload wordt met het HTTP PUT protocol gerealiseerd. Voor de upload moet er een uniek mediafile_id opgegeven worden die voor zowel de bestandsnaam op de SAN/NAS als de database entry wordt gebruikt.

6.4.1 Request URL

/mediafile/{mediafile_id}/upload [PUT]

6.4.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
-----------	--------	--------------	---------	---------------------

mediafile_id	alfanumeriek	ID van de mediafile die gevuld of overschreven moet worden	URL	verplicht
user_id	string	user id	GET	verplicht
filename	string	de naam van het originele bestand	GET	verplicht
retranscode	boolean	indien 'retranscode' niet gezet is, zullen alle bestaande transcodes van de originele mediafile verloren gaan. Als 'retranscode' 'true' is, dan zullen er nieuwe transcode jobs aangemaakt worden, die de bestaande transcodes vervangen	GET	optioneel; default = 'false'
create_still	boolean	Specificeer 'true' wanneer je een still gegenereerd wilt hebben na de upload en analyse.	GET	optioneel; default = 'false'

Voorbeeld request URL (PUT)

/mediafile/6c704fbd4ef58f2447fd1a3e7/upload?user_id=entree004&filename=video.wmv

6.4.3 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

6.5 FTP

Een andere mogelijkheid om bestanden te laten uploaden is via FTP. De ega kan zelf ftp-accounts aanmaken en beheren binnen zijn eigen applicatie. Na activatie kan een gebruiker dan zelf bestanden uploaden naar de server. De bestanden moeten tezamen met een identificerend XML bestand worden geüpload zodat VP-Core weet wat hiermee moet gebeuren.

6.6 Opvragen FTP gebruikers

Middels deze aanroep kan een EGA een lijst opvragen van de eigen FTP gebruikers.

6.6.1 Request URL

/ftp [GET]

6.6.2 Request parameters

geen.

6.6.3 Voorbeeld request URL (GET)

/ftp

6.6.4 Response fields

Field	Description
userid	id van de gebruiker

Voorbeeld response:

```
<items>
  <item id="1">
    <userid>nogeentestuser1000</userid>
  </item>
  <item id="2">
    <userid>testuser001000</userid>
  </item>
</items>
```

6.7 Aanmaken van een FTP gebruiker

Middels deze aanroep kan een EGA een FTP gebruiker aanmaken. De id van de EGA wordt als 3-cijferige tekst aan de gebruikersnaam toegevoegd, tenzij deze er al achter staat. De gebruikersnaam en het wachtwoord moeten uit minimaal 8 tekens bestaan.

6.7.1 Request URL

/ftp/create [POST]

6.7.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
password	string	wachtwoord van de nieuwe gebruiker	POST	verplicht
user	string	username van de nieuwe gebruiker.	POST	verplicht

6.7.3 Voorbeeld request URL

```
/ftp/create
POST:
password=secret123
user=hugo
```

6.7.4 Response

Field	Description
<request_result_id>601</request_result_id>	601: success

6.8 Opvragen van een FTP gebruiker

Met deze call kun je gegevens van een FTP gebruiker opvragen.

6.8.1 Request URL

/ftp/{user} [GET]

6.8.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user	string	username van de gebruiker.	URL	verplicht

6.9 FTP gebruiker wijzigen

Middels deze aanroep kan een EGA een FTP gebruiker het wachtwoord of de active status veranderen. Het wachtwoord moeten uit minimaal 8 tekens bestaan. Met de active setting kan de FTP gebruiker worden geblokkeert of weer geactiveerd worden.

6.9.1 Request URL

/ftp/{user}/update [POST]

6.9.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user	string	username van de gebruiker.	URL	verplicht
password	string	wachtwoord van de gebruiker	POST	verplicht tenzij 'active' niet gezet
active	boolean	account van de gebruiker actief?	POST	verplicht tenzij 'password' niet gezet

Voorbeeld request URL (POST)

```
/ftp/hugo001/update
POST:
password=secret456
active=true
```

6.10 Verwijderen van een FTP gebruiker

Middels deze aanroep kan een EGA een FTP gebruiker verwijderen.

6.10.1 Request URL

/ftp/{user}/delete [POST]

6.10.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user	string	username van de gebruiker.	URL	verplicht

6.10.3 Voorbeeld request:

/ftp/hugo001/delete

7 Transcoding

Transcoderen is het converteren van videobestanden van het ene formaat naar het andere. VP-Core ondersteunt Mpeg, WMV en Flash. VP-Core kan binnen deze formaten een grote variëteit aan parameters aan. Alleen originele mediafiles kunnen worden getranscodeerd.

Voor het transcoderen zijn drie mogelijkheden beschikbaar. Als er geen profile_id of tool wordt verstuurd, dan wordt een transcoding gestart met een default profiel. Andere parameters worden genegeerd. Wanneer er een profile_id wordt meegestuurd, dan zal via dat profiel worden getranscodeerd. Andere parameters worden genegeerd. Als er een tool is opgegeven dan wordt ook een file_extension verwacht. Een command is dan optioneel.

7.1 Transcoderen naar een default profiel

Deze aanroep implementeert het transcoderen van het bronbestand naar een default profiel. In de aanroep wordt het mediafile_id meegegeven. In de response wordt het job_id teruggegeven, deze kan gebruikt worden om bijvoorbeeld de status op te vragen. Er mag geen profile_id of tool worden opgegeven.

7.1.1 Request URL

/mediafile/{mediafile_id}/transcode [POST]

7.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	mediafile id	URL	verplicht
user_id	string	user id	POST	verplicht

Voorbeeld request URL (POST)

```
/mediafile/6c704fbd4ef58f2447fd1a3e7/transcode
POST:
user_id=entree004
```

7.1.3 Response fields

Field	Description
job_id	job id

7.1.4 Voorbeeld response

```
<item id="1">
  <job_id>7599</job_id>
</item>
```

7.2 Transcoderen naar een gekozen profiel

Deze aanroep implementeert het transcoderen van het bronbestand naar een gekozen profiel. In de aanroep wordt het mediafile_id en profiel_id meegegeven. In de response wordt het job_id teruggegeven, deze kan gebruikt worden om bijvoorbeeld de status op te vragen.

7.2.1 Request URL

/mediafile/{mediafile_id}/transcode [POST]

7.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	mediafile id	URL	verplicht
user_id	string	user id	POST	verplicht
profile_id	integer	profiel id	POST	optioneel

7.2.3 Voorbeeld request URL

/video/6c704fbd4ef58f2447fd1a3e7/transcode

POST:

user_id=entree004

profile_id=4

7.2.4 Response fields

Field	Description
job_id	job id

7.2.5 Voorbeeld response

```
<item id="1">
  <job_id>7599</job_id>
</item>
```

7.3 Transcoderen met behulp van opgegeven parameters

Deze aanroep implementeert het transcoderen van het bronbestand met behulp van opgegeven parameters. In de aanroep wordt het mediafile_id en de parameters meegegeven. In de response wordt het job_id teruggegeven, deze kan gebruikt worden om bijvoorbeeld de status op te vragen.

7.3.1 Request URL

/mediafile/{mediafile_id}/transcode [POST]

7.3.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	mediafile_id	URL	verplicht
user_id	string	user id	POST	verplicht
tool	alfanumeriek	naam van de gebruikte tool	POST	optioneel
file_extension	alfanumeriek	extensie van het bestand	POST	optioneel

command	string	een reeks van parameters	POST	optioneel
---------	--------	--------------------------	------	-----------

Wanneer er een profile_id is meegegeven dan wordt getranscodeerd volgens die profiel en worden alle andere parameters gedelete. Wanneer specifieke command parameters worden meegegeven dan moeten die op de volgende wijze worden opgebouwd:

key1:value1;key2:value2;

De mogelijke waarden voor key en value zijn:

Key	min	max	allowed	default
videobitrate	100	1024		200
fps	10	30		25
size			sqcif, qcif, cif, 4cif, qqvga, qvvga, vgasqcif, 128x96, qcif, 176x144, cif, 352x288, 4cif, 704x576, qqvga, 160x120, qvga, 320x240, vga, 640x480, svga, 800x600, xga, 1024x768, uxga, 1600x1200, qxga, 2048x1536, sxga, 1280x1024, qsxga, 2560x2048, hsxga, 5120x4096, wvga, 852x480, wxga, 1366x768, wsxga, 1600x1024, wuxga, 1920x1200, woxga, 2560x1600, wqsxga, 3200x2048, wquxga, 3840x2400, whsxga, 6400x4096, whuxga, 7680x4800, cga, 320x200, ega, 640x350, hd480, 852x480, hd720, 1280x720, hd1080, 1920x1080	qcif
qmax	1	31		
videocodec			flv, wmv, mpg	
audiocodec			mp3, wma	
audiobitrate			40, 64, 128, 192, 256	64
audiosamplingrate			22050, 44100	44100
audiochannels	0	2		1
duration	0			
startposition	0			

Voorbeeld request URL (POST)

/mediafile/6c704fbd4ef58f2447fd1a3e7/transcode

POST:

```

tool=ffmpeg
file_extension=flv
command=videobitrate:512;fps:30
user_id=entree004

```

7.3.3 Response fields

Field	Description
job_id	job id

7.3.4 Voorbeeld response

```

<items>
  <item id="1">
    <job_id>7596</job_id>
  </item>
</items>

```

7.4 Genereren van een still

Deze aanroep implementeert het genereren van een still op basis van het bronbestand met behulp van opgegeven parameters. In de aanroep wordt het mediafile_id en de parameters meegegeven. In de response wordt het job id teruggegeven, deze kan gebruikt worden om bijvoorbeeld de status op te vragen.

7.4.1 Request URL

/asset/{asset_id}/still/create [POST]

7.4.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	asset id	URL	verplicht
mediafile_id	alfanumeriek	mediafile id	POST	verplicht
user_id	string	user id	POST	verplicht

Voorbeeld request URL (POST)

```

/asset/6c704fbd4ef58f2447fd1a3e7/still/create
POST:
user_id=entree004
frametime=12

```

7.4.3 Response fields

Field	Description
job_id	job id

7.4.4 Voorbeeld response

```
<items>
  <item id="1">
    <job_id>7596</job_id>
  </item>
</items>
```

7.5 Opvragen van jouw default (preview) transcode profiel

Deze aanroep implementeert het opvragen van de door VP-Core ingestelde transcode profiel voor jouw applicatie.

7.5.1 Request URL

/preview_profile_id [GET]

7.5.2 Response fields

Field	Description
preview_profile_id	het profile id

7.5.3 Voorbeeld response

```
<items>
  <item id="1">
    <preview_profile_id>1</preview_profile_id>
  </item>
</items>
```

8 Jobs

8.1 Aanmaken van een transcodejob

Zie hoofdstuk 7

8.2 Ophalen van de transcodeprofielen

Deze aanroep implementeert het ophalen van alle transcodeprofielen die beschikbaar zijn. In de response wordt een lijst gegeven van de profielen. Een profielnaam kan daarna meegegeven worden aan een transcode job.

8.2.1 Request URL

/transcode/profiles [GET]

8.2.2 Request parameters

Geen

8.2.3 Voorbeeld request URL

/transcode/profiles

8.2.4 Voorbeeld response

```
<items>
  <item id="1">
    <profile_id>1</profile_id>
    <profile>standaard flash</profile>
  </item>
  <item id="2">
    <profile_id>2</profile_id>
    <profile>qcif flash</profile>
  </item>
  <item id="3">
    <profile_id>4</profile_id>
    <profile>standaard wmv</profile>
  </item>
  <item id="4">
    <profile_id>5</profile_id>
    <profile>standaard mp4</profile>
  </item>
  <item id="5">
    <profile_id>7</profile_id>
    <profile>FLASH 800x600</profile>
  </item>
</items>
```

8.2.5 Response fields

Field	Description
-------	-------------

profile_id	numerieke id van een profiel
profile	profielnaam

8.3 Ophalen van een lijst van jobs voor een asset

Deze aanroep implementeert het ophalen van een alle jobs die voor een asset zijn gedefinieerd. In de aanroep wordt het assetid en de parameters meegegeven. In de response wordt een lijst van jobs teruggegeven.

8.3.1 Request URL

/asset/{asset_id}/joblist [GET]

8.3.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	asset id	URL	verplicht
user_id	string	user id	GET	verplicht

Voorbeeld request URL (GET)

/asset/6c704fbd4ef58f2447fd1a3e7/joblist?user_id=entree004

8.3.3 Response fields

Field	Description
id	job id
owner	eigenaar van de job
status	status van de job (WAITING, IN PROGRESS, FINISHED, CANCELLED, FAILED)
progress	voortgang van de job
priority	prioriteit van de job
job_type	type job: TRANSCODE, UPLOAD, STILL, ANALYSE.
started	startdatum van de job, bijv: 28-10-2007 22:04
started_unix	starttijd van de job in een unix_timestamp formaat.
error_description	eventuele foutmelding

8.4 Ophalen van een lijst van jobs voor een user

Deze aanroep implementeert het ophalen van een alle jobs die voor een user zijn gedefinieerd. In de aanroep wordt het user_id en de parameters meegegeven. In de response wordt een lijst van jobs teruggegeven.

8.4.1 Request URL

/user/{owner_id}/joblist

8.4.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
owner_id	alfanumeriek	eigenaar van de job	URL	verplicht

8.4.3 Voorbeeld request URL

/user/entree004/joblist

8.4.4 Voorbeeld response

```
<item id="1">
  <id>209</id>
  <owner>entree004</owner>
  <status>FINISHED</status>
  <progress>1.000</progress>
  <priority>-1</priority>
  <job_type>TRANSCODE</job_type>
  <started>16-04-2008 00:20</started>
  <started_unix>1208298042</started_unix>
  <error_description></error_description>
</item>
<item id="2">
  <id>7589</id>
  <owner>entree004</owner>
  <status>INPROGRESS</status>
  <progress></progress>
  <priority>0</priority>
  <job_type>TRANSCODE</job_type>
  <started>18-04-2008 09:09</started>
  <started_unix>1208502587</started_unix>
  <error_description></error_description>
</item>
```

8.4.5 Response fields

Field	Description
id	job id
owner	eigenaar van de job
status	status van de job (WAITING, INPROGRESS, FINISHED, CANCELLED, FAILED)
progress	voortgang van de job
priority	prioriteit van de job
job_type	type job: TRANSCODE, UPLOAD, STILL, ANALYSE.

started	startdatum van de job, bijv: 28-10-2007 22:04
started_unix	starttijd van de job in een unix_timestamp formaat.
error_description	eventuele foutmelding

8.5 Opvragen van jobstatus

Deze aanroep implementeert het ophalen van de status van een job op basis van de opgegeven job id.

8.5.1 Request URL

/job/{job_id}/status

8.5.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
job_id	integer	job id	URL	verplicht
user_id	string	user id	GET	verplicht

Voorbeeld request URL (GET)

/job/34/status?user_id=entree004

8.5.3 Response fields

Field	Description
id	job id
owner	eigenaar van de job
status	status van de job (WAITING, INPROGRESS, FINISHED, CANCELLED, FAILED)
progress	voortgang van de job
priority	prioriteit van de job
job_type	type job: TRANSCODE, UPLOAD, STILL, ANALYSE.
started	startdatum van de job, bijv: 28-10-2007 22:04
started_unix	starttijd van de job in een unix_timestamp formaat.
error_description	eventuele foutmelding

8.6 Verwijderen van een job

Deze aanroep implementeert het verwijderen van een job op basis van de opgegeven job id. Een job kan alleen verwijderd worden wanneer deze de status 'WAITING', 'FINISHED' of 'FAILED' heeft.

8.6.1 Request URL

/job/{job_id}/delete [POST]

8.6.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
job_id	integer	job id	URL	verplicht
user_id	string	user id	POST	verplicht

8.6.3 Voorbeeld request URL

/job/34/delete

POST:

user_id=entree004

9 Assets

9.1 Lijst van assets ophalen

Deze aanroep geeft een lijst terug van alle beschikbare assets die aan de eventuele zoekopdracht voldoen.

9.1.1 Request URL

/asset [GET]

9.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	maximum aantal geretourneerde items	GET	verplicht
offset	numeriek	geef <limit> items terug vanaf <offset>	GET	optioneel
cql	string	Een CQL zoek instructie waarbij complexe zoek methodes worden ondersteund. Dit veld is exclusief met alle zoek- en order_* parameters. OF cql is gezet of de zoek- en order_* parameters. Indien beide worden gezet zal er een error worden teruggegeven. Voor meer informatie over de werking van CQL; zie document onder TO, TO CQL.doc of hoofdstuk in dit document 'CQL - Contextual Query Language'	GET	optioneel
order_by	string	Sorteer de lijst op een veld bv: order_by=changed Mogelijke sorteer velden: group_id, owner_id, provider_id, reference_id, changed, numofviews, en de dublin core velden en de qualified dublin core velden (zie hieronder), en de metadata_tags (zie desbetreffende paragraaf)	GET	optioneel
order_direction	string	Oplopend ('asc') of aflopend ('desc') sorteren. Voorbeeld: order_direction=desc	GET	optioneel
granted	bool	De granted bool is een switch waarbij de toegang het resultaat bepaald. Bij een setting false worden alleen de assets terug gegeven waar de gebruiker toegang tot heeft. Bij de waarde 'true' worden altijd alle assets terug gegeven, ongeacht of de gebruiker toegang heeft. In asset informatie in het resultaat staat een extra veld met de naam 'granted', deze bevat per asset of de gebruiker toegang had. In beide settings wordt deze 'granted' veld terug gegeven.	GET	optioneel

		De default waarde voor granted is 'true'.		
is_public_list	bool	'is_public_list' bepaald of de lijst voor iedereen te zien is. Deze setting heeft te maken met de isprivate setting van een asset. Indien op een asset isprivate waar is (TRUE) zal bij een setting van 'is_public_list' = true deze asset niet met de resultaten terug komen. Zelfs al is de gebruiker de owner van de asset waar isprivate = 'TRUE' is, zal deze niet terug komen bij 'is_public_list' = true. Bij de 'false' setting van 'is_public_list' wordt de waarde van 'isprivate' genegeerd.	GET	optioneel (default is false)
hide_empty_assets	bool	'hide_empty_assets' bepaald of alleen assets in het resultaat moeten komen die ook mediafile bevaten. Met de waarde 'true' zal alle assets overslaan die geen mediafiles bevaten.	GET	optioneel (default is false)
asset_id, group_id, owner_id, provider_id, reference_id, mediafile_duration, mediafile_container_type, filename, mime_type	string	Al deze velden kunnen meerdere malen gebruikt worden voor een zoekopdracht. voorbeelden: owner_id=test&group_id=test2 owner_id[]=test&owner_id[]=t2	GET	optioneel
zoeken op dublin core velden: type, language, title, creator, description, date, format, publisher, subject, contributor, identifier, source, relation, coverage_temporal, coverage_spatial, rights	string	Al deze velden kunnen meerdere malen gebruikt worden voor een zoekopdracht. voorbeelden: - subject=test&title=test2 - title[]=test&title[]=test2		optioneel
zoeken op qualified dublin core velden: isreferencedby, title_alternative, description_abstract, created, issued, hasformat, isformatof, format_medium, format_extent,	string	Al deze velden kunnen meerdere malen gebruikt worden voor een zoekopdracht.		optioneel

license, rightsholder				
...	string	eventuele metadata_tags toegevoegd met metadata_tag/create	GET	optioneel
<veldnaam>_match	string	Standaard zijn alle zoek matches 'contains' maar deze kan per veld aangepast worden dmv <veldnaam>_match=<type>. Keuze uit: 'exact', 'contains' en 'begin' Voorbeeld: title=test&title_match=exact	GET	optioneel
operator	operator	Voer de zoekopdracht uit met “and” of “or”. Voorbeeld: operator=or	GET	optioneel; standaard waarde = 'and'
videotimestamp, videotimestampmodified (de zoek syntax geldt voor alle datum velden: date (dc) created (qdc) issued (qdc) changed (asset))	ISO 8601 datum/tijd periode	Dit maakt het mogelijk binnen een tijdsperiode te zoeken. Het formaat voor een datetime is JJJJ-MM-DD HH:MM:SS. Waarbij alleen het jaartal verplicht is. Als je van 2006-2008 resultaat wilt zien wanneer assets waren aangemaakt doe je: videotimestamp=2006 2009 Het teken scheid de 2 datetimes. De 2e datetime wordt beschouwd als 'tot'. In het voorbeeld wordt dus alleen resultaat van 2006, 2007, 2008 terug gegeven. videotimestamp=2007-04-01 00:00:00 2007-05 Dit voorbeeld geeft alle resultaat van de maand april in 2007. Hetzelfde resultaat bereik je met videotimestamp=2007-04 2007-05	GET	optioneel
coll_id, coll_id[]	alfanumeriek	Geef alleen assets weer die in deze collectie(s) voorkomt. Je kunt in meerdere collecties zoeken door meerdere coll_id te specificeren, gebruik hiervoor [].	GET	optioneel
user_id, aut_user_id	string	De huidige gebruiker, wordt gebruikt voor autorisatie. De aut_* velden hebben invloed op het [granted] veld in de output, en indien de parameter “granted” is meegegeven het aantal records.	GET	optioneel
aut_group_id, aut_group_id[]	string	De groep(en) waar de huidige gebruiker zich in bevindt, wordt gebruikt voor autorisatie.	GET	optioneel
aut_domain	string	Het domain van de gebruiker, wordt gebruikt voor autorisatie.	GET	optioneel
aut_realm	string	Het realm van de gebruiker, wordt gebruikt voor autorisatie.	GET	optioneel
is_app_admin	boolean	Geeft aan of de rest call door een beheerder	GET	optioneel;

		wordt opgeroepen. Deze flag zorgt ervoor dat de beheerder 'niet gepaste content' kan zien.		default='false'
fav_user_id	string	Indien de fav_user_id wordt meegegeven zal alleen binnen de favorieten van deze gebruiker worden gezocht. Indien er verder geen zoekparameters worden gegeven, zal deze functie alle assets terug geven die favorite zijn van deze gebruiker.	GET	optioneel

9.1.3 Voorbeeld request URL

/asset?limit=200&order_by=created&order_direction=desc&title[]=test
 /asset?limit=2&offset=3

9.1.4 Voorbeeld response

```
<items>
  <item id="1">
    <asset_id>10061</asset_id>
    <app_id>1</app_id>
    <provider_id></provider_id>
    <owner_id>av-user-naa</owner_id>
    <group_id>beeldengeluid</group_id>
    <videotimestamp>2004-09-10 22:22:08</videotimestamp>
    <videotimestampmodified>2004-09-10 22:22:08</videotimestampmodified>
    <play_restriction_start></play_restriction_start>
    <play_restriction_end></play_restriction_end>
    <locked>FALSE</locked>
    <reference_id></reference_id>
    <isprivate>FALSE</isprivate>
    <is_protected>FALSE</is_protected>
    <dublin_core>
      <type>Video</type>
      <language>EN</language>
      <title>kurkentrekker.mpg</title>
      <creator>naa</creator>
      <publisher>naa</publisher>
      <description>this kurkentrekker.mpg of naa was n/naa/mpeg/kurkentrekker.mpg</description>
      <date>2004-09-10 22:22:08</date>
      <format></format>
      <subject></subject>
      <contributor></contributor>
      <identifier></identifier>
      <source></source>
      <relation></relation>
      <coverage_temporal></coverage_temporal>
      <coverage_spatial></coverage_spatial>
      <rights></rights>
    </dublin_core>
    <qualified_dublin_core>
      <title_alternative></title_alternative>
      <description_abstract></description_abstract>
      <created></created>
    </qualified_dublin_core>
  </item>
</items>
```

```

<issued></issued>
<hasformat></hasformat>
<isformatof></isformatof>
<format_medium></format_medium>
<format_extent></format_extent>
<license></license>
<rightsholder></rightsholder>
<isreferencedby></isreferencedby>
</qualified_dublin_core>
</item>
<item id="2">
  <asset_id>10062</asset_id>
  <app_id>1</app_id>
  <provider_id></provider_id>
  <owner_id>av-user-naa</owner_id>
  <group_id>beeldengeluid</group_id>
  <videotimestamp>2004-09-10 22:22:08</videotimestamp>
  <videotimestampmodified></videotimestampmodified>
  <play_restriction_start></play_restriction_start>
  <play_restriction_end></play_restriction_end>
  <locked>FALSE</locked>
  <reference_id></reference_id>
  <isprivate>FALSE</isprivate>
  <is_protected>FALSE</is_protected>
  <created>2008-04-07 09:53:44</created>
  <changed>2008-04-17 23:44:44</changed>
  <dublin_core>
    <type>Video</type>
    <language>EN</language>
    <title>interculturele-mpeg1-vhs.mpg</title>
    <creator>naa</creator>
    <publisher>naa</publisher>
    <description>this interculturele-mpeg1-vhs.mpg of naa was n/naa/mpeg/interculturele-mpeg1-vhs.mpg</description>
    <date>2004-09-10 22:22:08</date>
    <format></format>
    <subject></subject>
    <contributor></contributor>
    <identifier></identifier>
    <source></source>
    <relation></relation>
    <coverage_temporal></coverage_temporal>
    <coverage_spatial></coverage_spatial>
    <rights></rights>
  </dublin_core>
  <qualified_dublin_core>
    <title_alternative></title_alternative>
    <description_abstract></description_abstract>
    <created></created>
    <issued></issued>
    <hasformat></hasformat>
    <isformatof></isformatof>
    <format_medium></format_medium>
    <format_extent></format_extent>
    <license></license>
    <rightsholder></rightsholder>

```

```

<isreferencedby></isreferencedby>
</qualified_dublin_core>
</item>
</items>

```

9.1.5 Response fields

Field	Description
items	Volledige beschrijving van de assets incl. metadata in XML-formaat

9.2 Details van een enkele asset opvragen

Deze aanroep geeft de details terug van de asset inclusief metadata.

9.2.1 Request URL

/asset/{asset_id} [GET]

9.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset waarvan de informatie moet worden weergegeven	URL	verplicht
user_id, aut_user_id	string	De huidige gebruiker, wordt gebruikt voor autorisatie. De aut_* velden hebben invloed op het [granted] veld in de output, en indien de parameter "granted" is meegegeven het aantal records.	GET	optioneel
aut_group_id, aut_group_id[]	string	De groep(en) waar de huidige gebruiker zich in bevindt, wordt gebruikt voor autorisatie.	GET	optioneel
aut_domain	string	Het domain van de gebruiker, wordt gebruikt voor autorisatie.	GET	optioneel
aut_realm	string	Het realm van de gebruiker, wordt gebruikt voor autorisatie.	GET	optioneel

Voorbeeld request URL

/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z

9.2.3 Response fields

Field	Description
item	Volledige beschrijving van de asset incl. metadata in het gebruikelijke XML-formaat

9.3 Lijst van collections waarin de asset voor komt opvragen

Deze aanroep geeft een lijst van collections inclusief titel en beschrijving terug waarin de asset voorkomt.

9.3.1 Request URL

/asset/{asset_id}/collection [GET]

9.3.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset waarvan de informatie moet worden weergegeven	URL	verplicht
limit	numeriek	maximum aantal geretourneerde items	GET	verplicht
offset	numeriek	geef <limit> items terug vanaf <offset>	GET	optioneel
order_by	string	Sorteer de lijst op een veld bv: order_by=changed Mogelijke sorteer velden: title, description, owner_id, group_id, created, changed, private, public, category, numofvideos	GET	optioneel
order_direction	string	Oplopend ('asc') of aflopend ('desc') sorteren. Voorbeeld: order_direction=desc	GET	optioneel
created, changed	ISO 8601 datum/tijd periode	Dit maakt het mogelijk binnen een tijdsperiode te zoeken. Het formaat voor een datetime is JJJJ-MM-DD HH:MM:SS. Waarbij alleen het jaartal verplicht is. Als je van 2006-2008 resultaat wilt zien wanneer collections waren aangemaakt doe je: created=2006 2009 Het teken scheid de 2 datetimes. De 2e datetime wordt beschouwd als 'tot'. In het voorbeeld wordt dus alleen resultaat van 2006, 2007, 2008 terug gegeven. created=2007-04-01 00:00:00 2007-05 Dit voorbeeld geeft alle resultaat van de maand april in 2007. Hetzelfde resultaat bereik je met created=2007-04 2007-05	GET	optioneel
title, description, owner_id, group_id	string	Al deze velden kunnen meerdere malen gebruikt worden voor een zoekopdracht. voorbeelden: owner_id=test&group_id=test2 owner_id[]=test&owner_id[]=t2	GET	optioneel
<veldnaam>_match	string	Standaard zijn alle zoek matches 'contains' maar deze kan per veld aangepast worden dmv <veldnaam>_match=<type>. Keuze uit: 'exact', 'contains' en 'begin'	GET	optioneel

		Voorbeeld: title=test&title_match=exact		
private, category, public	boolean	Zoekt op boolean (true / false) zoek velden. Voorbeeld: public=true (geef alle collecties die public op true hebben)	GET	optioneel

9.3.3 Voorbeeld request URL

/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/collection?limit=100

9.3.4 Response fields

Field	Description
item	Volledige beschrijving van de collecties in het gebruikelijke XML-formaat

9.4 Asset aanmaken

Deze aanroep maakt een lege asset aan die later gevuld kan worden met metadata, mediafiles, e.d.

9.4.1 Request URL

/asset/create [POST]

9.4.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user_id	string	dit wordt gebruikt om eigendom vast te leggen in de database	POST	verplicht
group_id	alfanumeriek	dit wordt gebruikt om eigendom vast te leggen in de database	POST	optioneel
reference_id	string	Dit is een eenmalig te zetten veld, dat gebruikt kan worden ter (externe) identificatie van de asset.	POST	optioneel
provider_id	alfanumeriek		POST	optioneel

9.4.3 Voorbeeld request URL

/asset/create

POST:

user_id=username

group_id=groupname

9.4.4 Response fields

Field	Description
asset_id	Het gereserveerde asset_id

9.5 Asset informatie aanpassen

Via deze aanroep kunnen eigenschappen van een asset aangepast worden.

9.5.1 Request URL

/asset/{asset_id} [POST]

9.5.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset waarvan de informatie moet worden aangepast	URL	verplicht
user_id	string	dit wordt gebruikt voor autorisatie.	POST	verplicht
play_restriction_start	ISO 8601 datum/tijd	JJJJ-MM-DD HH:MM:SS	POST	optioneel
play_restriction_end	ISO 8601 datum/tijd	JJJJ-MM-DD HH:MM:SS	POST	optioneel
isprivate	boolean	Als isprivate=true, dan wordt de asset niet getoond in publieke lijsten, zoals top 10, laatst geupload, enz.	POST	optioneel, default = false
owner_id	string	De mogelijke nieuwe owner van de asset.	POST	optioneel
group_id	string	De mogelijke nieuwe group van de asset.	POST	optioneel
is_app_admin	boolean	Geeft aan of de rest call door een beheerder wordt opgeroepen. Deze flag zorgt ervoor dat de beheerder gezien wordt als enige de 'is_unappropriate' en/of 'owner_id' en/of 'group_id' mag wijzigen.	POST	optioneel; default = false
is_unappropriate	boolean	Als is_unappropriate=true dan kan alleen de owner of app beheerder deze asset (en onderliggende mediafiles) zien. Alleen de app beheerder kan deze flag wijzigen.	POST	optioneel, default = false

Voorbeeld request URL

```
/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z
play_restriction_start="2007-12-05 08:30:00"
play_restriction_end="2007-12-31 10:00:00"
```

9.5.3 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

9.6 Eigen metadata tag definitie toevoegen

Via deze aanroep kan de EGA eigen metadata definities aanmaken. Deze worden dan in asset lijsten opgenomen onder de hoofd tag <app_{app_id}>. Na aanmaken kan er ook op gezocht worden.

9.6.1 Request URL

/metadata_tag/create [POST]

9.6.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
name	alfanumeriek + underscore _	De te aan te maken tag (max 32 karakters)	POST	verplicht
type	'char', 'int' of 'datetime'	Het type van de tag	POST	verplicht

Voorbeeld request URL

```
/metadata_tag/create
POST:
name=custom_tag_1
type=int
```

Een voorbeeld zoekopdracht bij assets wordt dan:

```
/asset?limit=10&custom_tag_1=4
```

9.6.3 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

9.7 Eigen metadata tag definitie verwijderen

Via deze aanroep kan de EGA eigen metadata definities verwijderen.

Verwijderen is alleen mogelijk indien er geen metadata bestaat van deze tag, of als de GET parameter delete=cascade meegegeven wordt.

9.7.1 Request URL

/metadata_tag/delete [POST]

9.7.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
name	alfanumeriek + underscore _	Het aan te maken tag (max 32 karakters)	POST	verplicht
delete	'cascade'	Indien ook alle reeds aangemaakt metadata van deze tag verwijderd moet worden	POST	optioneel

Voorbeeld request URL

/metadata_tag/delete

POST:

name=custom_tag_1

delete=cascade

9.8 Metatags opvragen

Via deze aanroep kan de EGA alle globale en eigen metadata definities opvragen.

9.8.1 Request URL

/metadata_tag [GET]

9.8.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	maximum aantal items	GET	optioneel
offset	numeriek	geef <limit> items terug vanaf <offset>	GET	optioneel
order_by	limit, offset, order_by, order_direction	Het veld waarop gesorteerd moet worden.	GET	optioneel
order_direction	'asc' of 'desc'	Oplopend (asc) of aflopen (desc) sorteren	GET	optioneel; standaard sorteer richting is 'asc'
include_default_definitions	bool	Geef ook de standaard definities terug, zoals dublin core, qualified dublin core of CZP	GET	optioneel; standaard waarde TRUE
name	alfanumeriek + underscore _	Een exacte naam van de metadata tag, gebruikt om het bestaan van een metadata tag te testen. De setting include_default_definitions heeft effect op het testen, beste is deze op TRUE te laten.	GET	optioneel

9.9 Metadata aan enkele asset toevoegen

Via deze aanroep kan de metadata van een asset gezet worden. Bij een aanpassing moeten ook alle oude waardes meegestuurd worden, anders komen deze te vervallen.

9.9.1 Request URL

/asset/{asset_id}/metadata [POST]

9.9.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset waarvan de informatie moet worden aangepast	URL	verplicht
user_id	string	wordt gebruikt voor autorisatie.	POST	verplicht
replace	bool	'TRUE' of 'FALSE'. Standaard is TRUE. Definieer 'FALSE' als je de meegegeven parameters wil toevoegen en de overige (bestaande) wilt laten staan. Let op: als een parameter al bestaat, wordt deze niet overschreven.	POST	optioneel
action	string	Geeft aan hoe de metadataproperties gebruikt moeten worden; mogelijke waarden: 'replace', 'append', 'update'. 'replace' is default. Deze parameter kan niet gebruikt worden in combinatie met de vorige parameter: replace. action=replace: alle bestaande metadataproperties worden verwijderd, en de nieuwe gegevens worden opgeslagen. action=append: alle bestaande metadataproperties blijven bestaan, en de nieuwe gegevens worden hiernaast opgeslagen. action=update: alleen de gegevens van de meegegeven property definities worden vervangen, niet meegestuurde properties blijven ongewijzigd.	POST	optioneel
type	string	dublin core	POST	optioneel
format	string	dublin core	POST	optioneel
language	ISO 639-1-codes	2-letterige taal code, dublin core	POST	optioneel
title	string	dublin core	POST	optioneel

creator	string	dublin core	POST	optioneel
publisher	string	dublin core	POST	optioneel
subject	string	dublin core	POST	optioneel
description	string	dublin core	POST	optioneel
contributor	string	dublin core	POST	optioneel
date	ISO 8601 datum/tijd	JJJJ-MM-DD HH:MM:SS, dublin core	POST	optioneel
identifier	string	dublin core	POST	optioneel
source	string	dublin core	POST	optioneel
relation	string	dublin core	POST	optioneel
coverage_temporal	string	dublin core	POST	optioneel
coverage_spatial	string	dublin core	POST	optioneel
rights	string	dublin core	POST	optioneel
isreferencedby	string	qualified dublin core	POST	optioneel
title_alternative	string	qualified dublin core	POST	optioneel
description_abstract	string	qualified dublin core	POST	optioneel
created	ISO 8601 datum/tijd	qualified dublin core	POST	optioneel
issued	ISO 8601 datum/tijd	qualified dublin core	POST	optioneel
hasformat	string	qualified dublin core	POST	optioneel
isformatof	string	qualified dublin core	POST	optioneel
format_medium	string	qualified dublin core	POST	optioneel
format_extent	string	qualified dublin core	POST	optioneel
license	string	qualified dublin core	POST	optioneel
rightsholder	string	qualified dublin core	POST	optioneel
...	string	eventuele metadata_tags toegevoegd met metadata_tag/create	POST	optioneel

9.9.3 Voorbeeld request URL

Deze call wijzigt alle metadata en zet title en subject:

```
/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/metadata?user_id=entree004
POST: title="Titel"
subject[]="Subject 1"
subject[]="Subject 2"
```

En deze call voegt een extra 'title' en 2 extra 'subjects' toe aan de metadata.

```
/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/metadata?user_id=entree004
POST: title="Titel"
subject[]="Subject 1"
subject[]="Subject 2"
replace=FALSE
```

Als alleen de title geupdate hoeft te worden kan de volgende call gebruikt worden:

```
/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/metadata?user_id=entree004
POST: title="Nieuwe Titel"
action="update"
```

9.9.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

9.10 Een enkele asset verwijderen

Via deze aanroep kan de asset verwijderd worden. Assets kunnen alleen verwijderd worden als er geen mediafiles meer aan gekoppeld zijn of als de parameter delete=cascade meegegeven wordt.

Bij het verwijderen van de asset worden ook de eventuele stills, metadata en koppelingen met collections verwijderd.

9.10.1 Request URL

```
/asset/{asset_id}/delete [POST]
```

9.10.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de te verwijderen asset	URL	verplicht
user_id	string	wordt gebruikt voor autorisatie.	POST	verplicht
delete	'cascade'	Gebruik deze parameter om ook alle onderliggende items te verwijderen	POST	optioneel

Voorbeeld request URL

```
/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/delete?delete=cascade
```

9.10.3 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

9.11 Meerdere assets tegelijkertijd verwijderen

Via deze aanroep kunnen meerdere assets in een enkele request verwijderd worden. Assets kunnen alleen verwijderd worden als er geen mediafiles meer aan gekoppeld zijn of als de parameter delete=cascade meegegeven wordt.

Bij het verwijderen van de asset worden ook de eventuele stills, metadata en koppelingen met collections verwijderd.

9.11.1 Request URL

/asset/delete?asset_id[]={asset_id} [POST]

9.11.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id[]	alfanumeriek	de te verwijderen asset	POST	verplicht
user_id	string	wordt gebruikt voor autorisatie.	POST	verplicht
delete	'cascade'	Gebruik deze parameter om ook alle onderliggende items te verwijderen	POST	optioneel

9.11.3 Voorbeeld request URL

/asset/delete

POST:

delete=cascade

asset_id[]={VIN2ZDKv9rpsfmBAWpvfiw1zQ6z

asset_id[]={hjk8768klh809jnhkIKJ

9.11.4 Response fields

Field	Description
asset_id	ID van de asset die verwijderd wordt
result	Het resultaat van de actie
result_id	Het resultaat ID van de actie
result_description	De beschrijving van het resultaat van de actie

9.12 Lijst van mediafiles behorende bij een enkele asset opvragen

Deze aanroep geeft de details van alle mediafiles behorende bij de asset terug.

9.12.1 Request URL

/asset/{asset_id}/mediafile [GET]

9.12.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset waarvan de informatie moet worden weergegeven	URL	verplicht

9.12.3 Voorbeeld request URL

/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/mediafile

9.12.4 Response fields

Field	Description
item	Volledige beschrijving van de mediafiles behorende bij een asset incl. technische metadata in het gebruikelijke XML-formaat

9.13 De still behorende bij een enkele asset verwijderen

Via deze oproep kan een still behorende bij een asset worden verwijderd.

9.13.1 Request URL

/asset/{asset_id}/still/delete [POST]

9.13.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset_id waarvan de still verwijderd moet worden.	URL	verplicht
user_id	string	user id	POST	verplicht

9.13.3 Voorbeeld request URL

/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/still/delete

POST:

user_id=n.achternaam

9.13.4 Response fields

Field	Description
-------	-------------

n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie
-----	---

9.14 *Het aantal assets tellen per applicatie, group en/of owner*

Via deze oproep kan opgevraagd worden hoeveel assets er bij een bepaalde group en of user horen.

Zonder parameters worden alle assets van de applicatie geteld. Met toevoeging van de parameters kan de zoekopdracht verder verfijnd worden.

Het totale aantal assets wordt in de header vermeld als 'item_count_total'.

9.14.1 *Request URL*

/asset/count [GET]

9.14.2 *Request parameters*

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
owner_id	string	eigenaar van de assets	GET	optioneel
group_id	string	group van de assets	GET	optioneel

9.14.3 *Voorbeeld request URL*

/asset/count?owner_id=t.gebruiker&group_id=groep_a

9.14.4 *Voorbeeld response*

<item_count_total>44446</item_count_total>

9.14.5 *Response fields*

Field	Description
item_count_total	het totaal aantal assets van deze zoekopdracht (header)

9.15 *Het aantal collections tellen waar de asset in voor komt*

Via deze oproep kan opgevraagd worden in hoeveel collections de asset opgenomen is.

Het totale aantal collections wordt in de header vermeld als 'item_count_total'.

9.15.1 *Request URL*

/asset/{asset_id}/collection/count [GET]

9.15.2 *Request parameters*

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	asset waarvan de collecties geteld moeten	GET	verplicht

		worden		
--	--	--------	--	--

9.15.3 Voorbeeld request URL

/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/collection/count

9.15.4 Voorbeeld response

<item_count_total>1</item_count_total>

9.15.5 Response fields

Field	Description
item_count_total	het totaal aantal collections van deze zoekopdracht

9.16 Supplement aanmaken

Deze aanroep maakt een lege supplement aan die later gevuld kan worden met willekeurige data of valide XML ten behoeve van OAI-PMH

9.16.1 Request URL

/asset/{asset_id}/supplement/create [POST]

9.16.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	De asset waaraan het supplement gekoppeld moet worden	URL	verplicht
user_id	string	dit wordt gebruikt om eigendom vast te leggen in de database	POST	verplicht
group_id	alfanumeriek	dit wordt gebruikt om eigendom vast te leggen in de database	POST	optioneel

9.16.3 Voorbeeld request URL

/asset/uhesdf823hjKhfsd/supplement/create?user_id=username&group_id=groupname

9.16.4 Voorbeeld response

```
<items>
  <item id="1">
    <supplement_id>ZoaHgyOz8NezzGykP0g0JblJq47</supplement_id>
  </item>
</items>
```

9.16.5 Response fields

Field	Description
supplement_id	Het gereserveerde supplement_id

9.17 Supplementele data opslaan

Via deze aanroep kan supplementele data van een asset opgeslagen worden. Een asset kan maximaal 16 supplements bevatten van 64 Kilobyte per supplement.

9.17.1 Request URL

/asset/{asset_id}/supplement/{supplement_id}/update [POST]

9.17.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset_id	URL	verplicht
supplement_id	alfanumeriek	uniek supplement id	URL	verplicht
user_id	alfanumeriek	eigenaar van de asset	POST	verplicht
supplement	vrij / valide XML zonder xml-header indien OAI-PMH	data dat als supplement opgeslagen moet worden	POST	verplicht
oai_prefix	string	OAI-PMH specifieke informatie	POST	verplicht indien OAI-PMH
oai_namespaceuri	string	OAI-PMH specifieke informatie	POST	verplicht indien OAI-PMH
oai_schemalocation	string	OAI-PMH specifieke informatie	POST	verplicht indien OAI-PMH

9.17.3 Voorbeeld request URL

/asset/uhesdf823hjkhsd/supplement/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/update

POST:

user_id=entree004

supplement=<data>

9.17.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

9.18 Supplementele data van een enkele asset opvragen

Via deze aanroep kunnen de supplements van een enkele asset worden opgevraagd. De data wordt base64 geencodeerd geretourneerd.

9.18.1 Request URL

/asset/{asset_id}/supplement [GET]

9.18.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset_id waarvoor de supplement data moet worden opgehaald	URL	verplicht

9.18.3 Voorbeeld request URL

/asset/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/supplement

9.18.4 Voorbeeld response

```
<items>
  <item id="1">
    <supplement_id>5o97J5ap9n7kDyuzwlQb3QGclav</supplement_id>
    <asset_id>EaFZOO1YPXZKHs1r60nTSGh25Om</asset_id>
    <oai_prefix></oai_prefix>
    <oai_namespaceuri></oai_namespaceuri>
    <oai_schemalocation></oai_schemalocation>
    <created>2008-04-18 11:34:03</created>
    <changed>2008-04-18 11:34:03</changed>
    <supplement_base64></supplement_base64>
  </item>
  <item id="2">
    <supplement_id>ZoaHgyOz8NezzGykP0g0JblJq47</supplement_id>
    <asset_id>EaFZOO1YPXZKHs1r60nTSGh25Om</asset_id>
    <oai_prefix></oai_prefix>
    <oai_namespaceuri></oai_namespaceuri>
    <oai_schemalocation></oai_schemalocation>
    <created>2008-04-18 11:33:19</created>
    <changed>2008-04-18 11:33:19</changed>
    <supplement_base64></supplement_base64>
  </item>
</items>
```

9.18.5 Response fields

Field	Description
supplement_id	uniek id van het supplement
asset_id	uniek id van het asset
oai_prefix	OAI-PMH specifieke informatie

oai_namespaceuri	OAI-PMH specifieke informatie
oai_schemalocation	OAI-PMH specifieke informatie
created	datum supplement aangemaakt
changed	datum supplement
supplement_base64	base64 geencodeerde supplementele data

9.19 Supplementele data verwijderen

Via deze aanroep kan een supplement worden verwijderd.

9.19.1 Request URL

/asset/{asset_id}/supplement/{supplement_id}/delete [POST]

9.19.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user_id	string	dit wordt gebruikt om te bepalen of de user genoeg rechten heeft om het supplement te verwijderen	POST	verplicht
asset_id	alfanumeriek	De asset id waar de supplement toe behoort	URL	verplicht
supplement_id	alfanumeriek	Het id van de supplement die moet worden verwijderd	URL	verplicht

9.19.3 Voorbeeld request URL

/asset/EaFZOO1YPXZKHs1r60nTSGh25Om/supplement/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/delete

POST:

user_id=test_user

9.19.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

9.20 Favoriet gebruikers asset / collection link toevoegen

Via de call kan een gebruiker zijn favorieten vast leggen. Dat kan dat zijn met een collectie of met een asset.

9.20.1 Request URL

/user/{fav_user_id}/favorites/create [POST]

9.20.2 Request parameters

Parameter	Value	Description	Method	Required/Optional
fav_user_id	string	naam van de gebruiker	URI	verplicht
fav_type	alphanumeriek	Type; ASSET of COLLECTION. Voorbeeld; fav_type=ASSET	POST	verplicht
fav_id	alfanumeriek	De ID van of de asset of de collectie.	POST	verplicht

9.20.3 Voorbeeld request URL

/user/user0003/favorites/create?fav_type=asset&fav_id=22006

9.20.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

9.21 Favoriet gebruikers asset / collection link verwijderen

Via de call kan een gebruiker een favoriet verwijderen uit zijn favorieten. Dat kan met een collectie of met een asset.

9.21.1 Request URL

/user/{fav_user_id}/favorites/delete [GET]

9.21.2 Request parameters

Parameter	Value	Description	Methode	Required/Optional
fav_user_id	string	naam van de gebruiker	URI	verplicht
fav_type	alphanumeriek	Type; ASSET of COLLECTION. Voorbeeld; fav_type=ASSET	GET	verplicht
fav_id	alfanumeriek	De ID van of de asset of de collectie.	GET	verplicht

9.21.3 Voorbeeld request URL

/user/user0003/favorites/delete?fav_type=asset&fav_id=22006

9.21.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

9.22 Favorieten gebruiker opvragen

Via de call kunnen alle favorieten van een bepaalde gebruiker worden opgevraagd van een bepaald type; asset of collection.

9.22.1 Request URL

/user/{fav_user_id}/favorites [GET]

9.22.2 Request parameters

Parameter	Value	Description	Method	Required/Optional
fav_user_id	string	naam van de gebruiker	URL	verplicht
fav_type	alphanumeriek	Type; ASSET of COLLECTION. Voorbeeld; fav_type=ASSET	GET	verplicht
fav_id	alfanumeriek	De ID van of de asset of de collectie.	GET	verplicht

9.23 Aantal keren asset in favorieten

Met deze call kun je opvragen hoe vaak een asset bij verschillende gebruikers in hun favorieten voorkomt.

9.23.1 Request URL

asset/{asset_id}/favorites/count [GET]

9.23.2 Request parameters

Parameter	Value	Description	Method	Required/Optional
asset_id	alfanumeriek	Asset id van de asset waar je een telling van wilt hebben hoe vaak deze in favorieten voorkomt.	URL	verplicht

9.24 Aantal keren collection in favorieten

Met deze call kun je opvragen hoe vaak een asset bij verschillende gebruikers in hun favorieten voorkomt.

9.24.1 Request URL

collection/{coll_id}/favorites/count [GET]

9.24.2 Request parameters

Parameter	Value	Description	Method	Required/Optional
coll_id	numeriek	De coll id van de collection waar je een telling van wilt hebben hoe vaak deze in favorieten voorkomt.	GET	verplicht

10 Collections

10.1 Lijst van collections opvragen

Deze aanroep geeft een lijst terug van alle beschikbare collecties die aan de eventuele zoekopdracht voldoen.

10.1.1 Request URL

/collection [GET]

10.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	maximum aantal items	GET	verplicht
offset	numeriek	geef <limit> items terug vanaf <offset>	GET	optioneel
cql	string	Een CQL zoek instructie waarbij complexe zoek methodes worden ondersteund. Dit veld is exclusief met alle zoek- en order_* parameters. OF cql is gezet of de zoek- en order_* parameters. Indien beide worden gezet zal er een error worden teruggeven. Voor meer informatie over de werking van CQL; zie document onder TO, TO CQL.doc of hoofdstuk in dit document 'CQL - Contextual Query Language'	GET	optioneel
user_id	string	De huidige gebruiker is nodig om deze gebruiker toegang te geven tot eigen collections waarbij isprivate = 'true' heeft.	GET	optioneel
is_app_admin	boolean	Geeft aan of de rest call door een beheerder wordt opgeroepen. Deze flag zorgt ervoor dat de beheerder 'niet gepaste content' kan zien.	GET	optioneel; default='false'
order_by	title, description, owner_id, group_id, created, changed of numofvideos	Het veld waarop gesorteerd moet worden. In het geval van numofvideos wordt standaard in de tegenovergestelde richting gesorteerd (desc), zodat, als er geen order_direction wordt aangegeven, de collecties met de meeste assets als eerst in volgorde wordt terug gegeven.	GET	optioneel
order_direction	'asc' of 'desc'	Oplopend (asc) of aflopen (desc) sorteren	GET	optioneel; standaard sorteer richting is 'asc'
title, description, owner_id, group_id	string	Al deze velden kunnen meerdere malen gebruikt worden voor een zoekopdracht. voorbeelden: .description=test&title=test2... .title[]=test&title[]=test2...	GET	optioneel
title_match, description_match,	string	Standaard zijn alle zoek matches 'contains' maar deze kan per veld aangepast worden dmv	GET	optioneel; standaard

owner_id_match, group_id_match		<veldnaam>_match=<type>. Keuze uit: 'exact', 'contains' en 'begin' Voorbeeld: title=test&title_match=exact		zoekoptie is 'contains'
private, public, category	boolean	Dit zijn boolean velden; true of false als waarde. Voorbeelden: private=false public=true	GET	optioneel
created, changed	ISO 8601 datum/tijd periode	Dit maakt het mogelijk binnen een tijdsperiode te zoeken. Het formaat voor een datetime is JJJJ-MM-DD HH:MM:SS. Waarbij alleen het jaartal verplicht is. Als je van 2006-2008 resultaat wilt zien wanneer collections waren aangemaakt doe je: created=2006 2009 Het teken scheid de 2 datetimes. De 2e datetime wordt beschouwd als 'tot'. In het voorbeeld wordt dus alleen resultaat van 2006, 2007, 2008 terug gegeven. created=2007-04-01 00:00:00 2007-05 Dit voorbeeld geeft alle resultaat van de maand april in 2007. Hetzelfde resultaat bereik je met created=2007-04 2007-05	GET	optioneel
operator	operator	Voer de zoekopdracht uit met "and" of "or". Voorbeeld: operator=or	GET	optioneel; standaard waarde = 'and'

10.1.3 Voorbeeld request URL

/collection?limit=2&order_by=title

10.1.4 Voorbeeld response

```
<items>
<item id="1">
  <coll_id>1</coll_id>
  <title>Fontys FB Media</title>
  <description>presentaties voor mediatheekmedewerkers</description>
  <app_id>1</app_id>
  <owner_id>admin-fontys1</owner_id>
  <group_id>fontys</group_id>
  <isprivate>FALSE</isprivate>
  <is_protected>FALSE</is_protected>
  <created>2008-04-07 09:53:44</created>
  <changed>2008-04-17 23:45:34</changed>
</item>
<item id="2">
```

```

<coll_id>10</coll_id>
<title>Terena Networking Conference 2003</title>
  <description>Collectie met presentaties die door SURFnet gehouden zijn tijdens de Terena Networking Conference 2003
(TNC2003) in Zagreb.</description>
  <app_id>1</app_id>
  <owner_id>roland</owner_id>
  <group_id>surfnet</group_id>
  <isprivate>FALSE</isprivate>
  <is_protected>FALSE</is_protected>
  <created>2008-04-07 09:53:44</created>
  <changed>2008-04-17 23:45:34</changed>
</item>
</items>
    
```

10.1.5 Response fields

Field	Description
item	Volledige beschrijving van de collecties incl. metadata in het gebruikelijke XML-formaat

10.2 Details van een enkele collection opvragen

Deze aanroep geeft de details terug van de collection inclusief metadata.

10.2.1 Request URL

/collection/{coll_id} [GET]

10.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
coll_id	alfanumeriek	de collectie waarvan de informatie moet worden weergegeven	URL	verplicht

10.2.3 Voorbeeld request URL

/collection/Ainqv6AN2GCUcO87G7AUvRnnv

10.2.4 Voorbeeld response

```

<items>
  <item id="1">
    <coll_id>Ainqv6AN2GCUcO87G7AUvRnnvhn</coll_id>
    <title>Frans Test Collectie</title>
    <description>Frans Test Collectie</description>
    <app_id>1</app_id>
    <owner_id>frans.ward</owner_id>
    <group_id>1</group_id>
    <isprivate>FALSE</isprivate>
    <is_protected>FALSE</is_protected>
    <created>2008-04-15 21:50:41</created>
  </item>
</items>
    
```

```

    <changed>2008-04-15 21:50:41</changed>
  </item>
</items>

```

10.2.5 Response fields

Field	Description
item	Volledige beschrijving van de collectie incl. metadata in het gebruikelijke XML-formaat

10.3 Lijst van assets behorende bij een enkele collection opvragen

Deze aanroep geeft alle assets terug die in de collection zitten inclusief metadata. Bij deze call kunnen dezelfde sorteer en zoek functies gebruikt worden als bij het opvragen van een asset lijst.

10.3.1 Request URL

/collection/{coll_id}/asset [GET]

10.3.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
coll_id	alfanumeriek	de collectie waarvan de informatie moet worden weergegeven	URL	verplicht
limit	numeriek	maximum aantal items	GET	verplicht
offset	numeriek	geef <limit> items terug vanaf <offset>	GET	optioneel
order_by	string	Sorteer de lijst op een veld bv: order_by=changed Mogelijke sorteer velden: group_id, owner_id, provider_id, reference_id, changed, en de dublin core velden en de qualified dublin core velden (zie hieronder), en de metadata_tags (zie desbetreffende paragraaf)	GET	optioneel
order_direction	string	Oplopend ('asc') of aflopend ('desc') sorteren. Voorbeeld: order_direction=desc	GET	optioneel
granted	bool	De granted bool is een switch waarbij de toegang het resultaat bepaald. Bij een setting false worden alleen de assets terug gegeven waar de gebruiker toegang tot heeft. Bij de waarde 'true' worden altijd alle assets terug gegeven, ongeacht of de gebruiker toegang heeft.	GET	optioneel

		In asset informatie in het resultaat staat een extra veld met de naam 'granted', deze bevat per asset of de gebruiker toegang had. In beide settings wordt deze 'granted' veld terug gegeven. De default waarde voor granted is 'true'.		
asset_id, group_id, owner_id, provider_id, reference_id, mediafile_duration, mediafile_container_type	string	Al deze velden kunnen meerdere malen gebruikt worden voor een zoekopdracht. voorbeelden: owner_id=test&group_id=test2 owner_id[]=test&owner_id[]=t2	GET	optioneel
zoeken op dublin core velden: type, language, title, creator, description, date, format, publisher, subject, contributor, identifier, source, relation, coverage_temporal, coverage_spatial, rights	string	Al deze velden kunnen meerdere malen gebruikt worden voor een zoekopdracht. voorbeelden: – subject=test&title=test2 – title[]=test&title[]=test2	GET	optioneel
zoeken op qualified dublin core velden: isreferencedby, title_alternative, description_abstract, created, issued, hasformat, isformatof, format_medium, format_extent, license, rightsholder	string	Al deze velden kunnen meerdere malen gebruikt worden voor een zoekopdracht.	GET	optioneel
...	string	eventuele metadata_tags toegevoegd met metadata_tag/create	GET	optioneel
<veldnaam>_match	string	Standaard zijn alle zoek matches 'contains' maar deze kan per veld aangepast worden dmv <veldnaam>_match=<type>. Keuze uit: 'exact', 'contains' en 'begin' Voorbeeld: title=test&title_match=exact	GET	optioneel
operator	operator	Voer de zoekopdracht uit met “and” of “or”.	GET	optioneel; standaard waarde = 'and'

		Voorbeeld: operator=or		
videotimestamp, videotimestampmodified (de zoek syntax geldt voor alle datum velden: date (dc) created (qdc) issued (qdc) changed (asset))	ISO 8601 datum/tijd periode	Dit maakt het mogelijk binnen een tijdsperiode te zoeken. Het formaat voor een datetime is JJJJ-MM-DD HH:MM:SS. Waarbij alleen het jaartal verplicht is. Als je van 2006-2008 resultaat wilt zien wanneer assets waren aangemaakt doe je: videotimestamp=2006 2009 Het teken scheid de 2 datetimes. De 2e datetime wordt beschouwd als 'tot'. In het voorbeeld wordt dus alleen resultaat van 2006, 2007, 2008 terug gegeven. videotimestamp=2007-04-01 00:00:00 2007-05 Dit voorbeeld geeft alle resultaat van de maand april in 2007. Hetzelfde resultaat bereik je met videotimestamp=2007-04 2007-05	GET	optioneel
is_app_admin	boolean	Geeft aan of de rest call door een beheerder wordt opgeroepen. Deze flag zorgt ervoor dat de beheerder 'niet gepaste content' kan zien.	GET	optioneel; default='false'
user_id, aut_user_id	string	De huidige gebruiker, wordt gebruikt voor autorisatie. De aut_* velden hebben invloed op het [granted] veld in de output, en indien de parameter "granted" is meegegeven het aantal records.	GET	optioneel
aut_group_id, aut_group_id[]	string	De groep(en) waar de huidige gebruiker zich in bevindt, wordt gebruikt voor autorisatie.	GET	optioneel
aut_domain	string	Het domain van de gebruiker, wordt gebruikt voor autorisatie.	GET	optioneel
aut_realm	string	Het realm van de gebruiker, wordt gebruikt voor autorisatie.	GET	optioneel

10.3.3 Voorbeeld request URL

/collection/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/asset?limit=200&order_by=created&order_direction=desc&title[]=test

10.3.4 Voorbeeld response

```
<item id="1">
  <asset_id>1</asset_id>
  <app_id>1</app_id>
  <provider_id></provider_id>
```

```

<owner_id>av-user-zfontys</owner_id>
<group_id>fontys</group_id>
<videotimestamp>2004-09-10 19:59:26</videotimestamp>
<videotimestampmodified></videotimestampmodified>
<play_restriction_start></play_restriction_start>
<play_restriction_end></play_restriction_end>
<locked>FALSE</locked>
<reference_id></reference_id>
<isprivate>FALSE</isprivate>
<dublin_core>
  <type>Video</type>
  <language>NL</language>
  <title>bijeenkomst mediatheken 10 maart</title>
  <creator>g.bierens</creator>
  <description>De personeelsbijeenkomst voor alle medewerkers van de mediatheken stond in het teken van "De
medewerker als partner in learning communities". Het doel was om de aanstaande veranderingen in de mediatheken in een
strategisch perspectief te zetten.</description>
  <date>2004-09-10 19:59:26</date>
  <format></format>
  <publisher></publisher>
  <subject></subject>
  <contributor></contributor>
  <identifier></identifier>
  <source></source>
  <relation></relation>
  <coverage_temporal></coverage_temporal>
  <coverage_spatial></coverage_spatial>
  <rights></rights>
</dublin_core>
<qualified_dublin_core>
  <title_alternative>De medewerker als partner in learning communities</title_alternative>
  <description_abstract></description_abstract>
  <created></created>
  <issued></issued>
  <hasformat></hasformat>
  <isformatof></isformatof>
  <format_medium></format_medium>
  <format_extent></format_extent>
  <license></license>
  <rightsholder></rightsholder>
  <isreferencedby></isreferencedby>
</qualified_dublin_core>
</item>
<item id="2">
  <asset_id>15745</asset_id>
  <app_id>1</app_id>
  <provider_id></provider_id>
  <owner_id>av-user-zfontys</owner_id>
  <group_id>fontys</group_id>
  <videotimestamp>2005-08-23 13:44:26</videotimestamp>
  <videotimestampmodified></videotimestampmodified>
  <play_restriction_start></play_restriction_start>
  <play_restriction_end></play_restriction_end>
  <locked>FALSE</locked>
  <reference_id></reference_id>

```

```

<isprivate>FALSE</isprivate>
<dublin_core>
  <type>Video</type>
  <language>NL</language>
  <title>Fontys Mediatheek TF Introductiefilm</title>
  <creator>Fokke M&#xFC;ller</creator>
  <creator>Erik Oomen</creator>
  <description>Mediatheek introductie</description>
  <date>2005-08-23 13:44:26</date>
  <format></format>
  <publisher></publisher>
  <subject></subject>
  <contributor></contributor>
  <identifier></identifier>
  <source></source>
  <relation></relation>
  <coverage_temporal></coverage_temporal>
  <coverage_spatial></coverage_spatial>
  <rights></rights>
</dublin_core>
<qualified_dublin_core>
  <title_alternative></title_alternative>
  <description_abstract></description_abstract>
  <created></created>
  <issued></issued>
  <hasformat></hasformat>
  <isformatof></isformatof>
  <format_medium></format_medium>
  <format_extent></format_extent>
  <license></license>
  <rightsholder></rightsholder>
  <isreferencedby></isreferencedby>
</qualified_dublin_core>
</item>
</items>

```

10.3.5 Response fields

Field	Description
item	Volledige beschrijving van de assets uit een collectie incl. metadata in het gebruikelijke XML-formaat

10.4 Collection aanmaken

Deze aanroep maakt een lege collectie aan die later gevuld kan worden met assets, metadata, e.d.

10.4.1 Request URL

/collection/create [POST]

10.4.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user_id	string	dit wordt gebruikt om eigendom vast te	POST	verplicht

		leggen in de database		
group_id	alfanumeriek	dit wordt gebruikt om eigendom vast te leggen in de database	POST	optioneel
private	'true' / 'false'	of het een 'private' collection betreft	POST	optioneel
public	'true' / 'false'	of het een 'public' collection betreft	POST	optioneel
category	'true' / 'false'	of het een 'category' collection betreft	POST	optioneel
isprivate	'true' / 'false'		POST	optioneel
public_assign	'true' / 'false'	Een 'true' waarde geeft alle gebruikers die toegang hebben tot de collectie het recht om assets, waarvan eigenaar, in de collectie te plaatsen of te verwijderen.	POST	optioneel

10.4.3 Voorbeeld request URL

/collection/create

POST:

user_id=frans.ward

group_id=wij

10.4.4 Voorbeeld response

```
<items>
  <item id="1">
    <coll_id>5nQPE1YDcYfzT2AEN1mLZMUgU3e</coll_id>
  </item>
</items>
```

10.4.5 Response fields

Field	Description
coll_id	het gereserveerde coll_id

10.5 Collection informatie aanpassen

Via deze aanroep kunnen eigenschappen van een collection aangepast worden.

10.5.1 Request URL

/collection/{coll_id} [POST]

10.5.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
-----------	--------	--------------	---------	---------------------

coll_id	alfanumeriek	de collection waarvan de informatie moet worden aangepast	URL	verplicht
user_id	string	dit wordt gebruikt om eigendom vast te leggen in de database	POST	verplicht
title	string	Titel van de collectie	POST	optioneel
description	string	Beschrijving van de collectie	POST	optioneel
private	'true' / 'false'	of het een 'private' collection betreft	POST	optioneel
public	'true' / 'false'	of het een 'public' collection betreft	POST	optioneel
category	'true' / 'false'	of het een 'category' collection betreft	POST	optioneel
isprivate	'true' / 'false'		POST	optioneel
public_assign	'true' / 'false'	Een 'true' waarde geeft alle gebruikers die toegang hebben tot de collectie het recht om assets, waarvan eigenaar, in de collectie te plaatsen of te verwijderen.	POST	optioneel
owner_id	string	De mogelijke nieuwe owner van de collection.	POST	optioneel
group_id	string	De mogelijke nieuwe group van de collection.	POST	optioneel
is_app_admin	boolean	Geeft aan of de rest call door een beheerder wordt opgeroepen. Deze flag zorgt ervoor dat de beheerder gezien wordt als enige de 'owner_id' en/of 'group_id' mag wijzigen.	POST	optioneel; default = false

10.5.3 Voorbeeld request URL

/collection/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z

POST:

title="Titel van de collectie"

description="Beschrijving van de collectie"

10.5.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

10.6 Een of meerdere assets aan een collectie koppelen

Via deze aanroep kunnen een of meerdere assets toegevoegd worden aan een collectie.

10.6.1 Request URL

/collection/{coll_id}/asset_relation [POST]

10.6.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
coll_id	numeriek	applicatie (EGA) id	URL	verplicht
asset_id of asset_id[]	alfanumeriek	de asset waarvan de informatie moet worden aangepast	POST	verplicht
user_id	string	dit wordt gebruikt om eigendom vast te leggen in de database	POST	verplicht

Voorbeeld request URL voor het koppelen van een enkele asset

/collection/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/asset_relation

POST:

asset_id=QeflsKE9RCUKDtvYfnxnJ3guo5h

user_id=test_user

Voorbeeld request URL voor het koppelen van meerdere enkele assets

/collection/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/asset_relation

POST:

asset_id[]=QeflsKE9RCUKDtvYfnxnJ3guo5h

asset_id[]=5w1kkY7PuRhItICqkPEv6fhL6vM

user_id=test_user

10.6.3 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

10.7 Een of meerdere asset/collectie koppelingen verwijderen

Via deze aanroep kunnen assets verwijderd worden uit een collectie.

10.7.1 Request URL

/collection/{coll_id}/asset_relation/delete [POST]

10.7.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
coll_id	numeriek	de collectie waar de	URL	verplicht

		asset uit gehaald moet worden		
user_id	string	dit wordt gebruikt voor autorisatie controle	POST	verplicht
asset_id[]	alfanumeriek	de asset die uit de collectie gehaald moet worden	POST	verplicht

10.7.3 Voorbeeld request URL

/collection/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/asset_relation/delete?
asset_id[]=QeflsKE9RCUKDtvYfnxnJ3guo5h&asset_id[]=5w1kkY7PuRhItlCqkPEv6fhL6vM

10.7.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

10.8 Collectie verwijderen

Via deze aanroep kan de collection verwijderd worden. Collections kunnen alleen verwijderd worden als er geen assets meer aan gekoppeld zijn of als de parameter delete=cascade meegegeven wordt.

Bij 'delete=cascade' worden ook alle gekoppelde assets en de daar aan gekoppelde mediafiles, stills en metadata verwijderd, ook al worden de assets ook nog in andere collections gebruikt. Per asset wordt wel gecontroleerd of de user_id ook de eigenaar van de asset is, anders faalt het verwijderen.

10.8.1 Request URL

/collection/{coll_id}/delete [POST]

10.8.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
coll_id	alfanumeriek	de te verwijderen collection	URL	verplicht
user_id	string	dit wordt gebruikt voor autorisatie controle	POST	verplicht
delete	'cascade'	Gebruik deze parameter om ook alle onderliggende items te verwijderen. De collectie wordt alleen verwijderd als ook alle onderliggende assets succesvol verwijderd zijn.	POST	optioneel

10.8.3 Voorbeeld request URL

/collection/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/delete

POST:

delete=cascade

10.8.4 Response fields

Field	Description
n/a	success of bij delete=cascade een lijst van verwijderde asset id's.

10.9 Het aantal collections tellen per applicatie, group en/of owner

Via deze oproep kan opgevraagd worden hoeveel collections er bij een bepaalde group en of user horen.

Zonder parameters worden alle collections van de applicatie geteld. Met toevoeging van de parameters word de zoekopdracht verfijnd.

Het totale aantal collections wordt in de header vermeld als 'item_count_total'.

10.9.1 Request URL

/collection/count [GET]

10.9.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
owner_id	string	eigenaar van de collections	GET	optioneel
group_id	string	group van de collections	GET	optioneel

10.10 Voorbeeld request URL

/collection/count?owner_id=t.gebruiker&group_id=groep_a

10.10.1 Response fields

Field	Description
item_count_total	Het totaal aantal collections van deze zoekopdracht.

11 Mediafiles

11.1 Mediafile aanmaken

Deze aanroep maakt een lege mediafile aan die later gebruikt kan worden voor een file-upload of voor het opslaan van een externe stream url.

Mediafiles waarvan 'uri' gezet is kunnen niet meer gebruikt worden om naartoe te uploaden. Mediafiles waar de 'filename' gezet is kunnen ook niet meer gebruikt worden om een URL in te zetten.

11.1.1 Request URL

mediafile/create [POST]

11.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
asset_id	alfanumeriek	de asset waaraan de mediafile moet worden gekoppeld	POST	verplicht
user_id	string	dit wordt gebruikt om eigendom vast te leggen in de database	POST	verplicht
group_id	alfanumeriek	dit wordt gebruikt om eigendom vast te leggen in de database	POST	optioneel
uri	string	de URL van een externe stream	POST	optioneel
is_downloadable	'true' of 'false'	of de mediafile gedownload mag worden	POST	optioneel (mag niet in combinatie met uri gebruikt worden) default = false.

11.1.3 Voorbeeld request URL

```
/mediafile/create?
POST:
asset_id=jeJbje9ujwer3
user_id=username
group_id=groupname
uri=rtsp://server.ext/stream
```

11.1.4 Voorbeeld response

```
<items>
  <item id="1">
    <mediafile_id>MeOxhYlHq97Db71EdxSBcehuhb</mediafile_id>
  </item>
</items>
```

11.1.5 Response fields

Field	Description
mediafile_id	het gereserveerde mediafile_id

11.2 Details van een enkele mediafile opvragen

Deze aanroep geeft de details van de gespecificeerde mediafile terug.

11.2.1 Request URL

/mediafile/{mediafile_id} [GET]

11.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	de mediafile waarvan de informatie moet worden weergegeven	URL	verplicht

11.2.3 Voorbeeld request URL

/mediafile/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z

11.2.4 Voorbeeld response

```
<items>
  <item id="1">
    <mediafile_id>8JrogQ8OqxM8kuSC8KCvIZJ8WYa</mediafile_id>
    <asset_id>2hdMYNeybzVxVQQSITrfMFBNW5c</asset_id>
    <app_id>1</app_id>
    <owner_id>entree004</owner_id>
    <group_id>wij</group_id>
    <is_original_file>TRUE</is_original_file>
    <is_downloadable>FALSE</is_downloadable>
    <filename></filename>
    <uri></uri>
    <transcode_profile_id></transcode_profile_id>
    <tool></tool>
    <command></command>
    <file_extension></file_extension>
    <is_protected>FALSE</is_protected>
    <created>2008-04-20 09:46:06</created>
    <changed>2008-04-20 09:46:06</changed>
    <metadata>
      <video_codec>wmv2</video_codec>
      <colorspace>yuv420p</colorspace>
      <width>208</width>
      <height>160</height>
      <fps>20.00</fps>
      <audio_codec>wmav2</audio_codec>
      <sample_rate>22050</sample_rate>
      <channels>stereo</channels>
    </metadata>
  </item>
</items>
```

```
<file_duration>00:05:14.5</file_duration>
<container_type>asf</container_type>
<bitrate>226</bitrate>
<bpp>0.34</bpp>
<filesize>8893674</filesize>
<created>2008-04-17 04:17:33</created>
<changed>2008-04-17 04:17:33</changed>
</metadata>
</item>
</items>
```

11.2.5 Response fields

Field	Description
item	Volledige beschrijving van de mediafiles behorende bij een asset incl. technische metadata in het gebruikelijke XML-formaat

11.3 Informatie van een enkele mediafile aanpassen

Via deze aanroep kunnen eigenschappen van de mediafile aangepast worden. De post waardes 'uri' en 'filename' mogen niet samen gebruikt worden.

11.3.1 Request URL

/mediafile/{mediafile_id} [POST]

11.3.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
mediafile_id	alfanumeriek	de mediafile waarvan de informatie moet worden weergegeven	URL	verplicht
user_id	string	dit wordt gebruikt om eigendom vast te leggen in de database	POST	verplicht
filename	string	de bestandsnaam die de mediafile weer aanneemt als hij gedownload wordt	POST	optioneel (mag niet in combinatie met uri gebruikt worden)
is_downloadable	'true' of 'false'	of de mediafile gedownload mag worden	POST	optioneel (mag niet in combinatie met uri gebruikt worden), default = false.
uri	string	de URL van een externe stream	POST	optioneel (mag niet in combinatie met is_downloadable of filename gebruikt worden)
transcode_inherits_acl	bool	Waarde 'true'; Bij toekomstige transcodes op dit orgineel worden alle mediafile acl rechten gecopieerd op de nieuwe	POST	optioneel; default = 'false'

		transcoded mediafile. Setting wordt alleen gebruikt indien de mediafile ook het origineel is. Het zetten van deze flag op niet-originele mediafiles heeft geen invloed op transcoded versie op het origineel.		
--	--	---	--	--

11.3.3 Voorbeeld request URL

/mediafile/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z
filename=naam.extensie

11.3.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

11.4 Een enkele mediafile verwijderen

Via deze aanroep kan een mediafile (origineel of transcode) verwijderd worden. Eventuele technische metadata en/of afschermingsregels worden ook verwijderd.

11.4.1 Request URL

/mediafile/{mediafile_id}/delete [POST]

11.4.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user_id	string	dit wordt gebruikt voor autorisatie.	POST	verplicht
mediafile_id	alfanumeriek	de te verwijderen mediafile	URL	verplicht

11.4.3 Voorbeeld request URL

/mediafile/VIN2ZDKv9rpsfmBAWpvfiw1zQ6z/delete
POST:
user_id=entree004

11.4.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

11.5 Het aantal mediafiles tellen per applicatie, group en/of owner

Via deze oproep kan opgevraagd worden hoeveel mediafiles er bij een bepaalde group en of user horen.

Zonder parameters worden alle mediafiles van de applicatie geteld. Met toevoeging van de parameters word de zoekopdracht verfijnd.

Het totale aantal mediafiles wordt in de header vermeld als 'item_count_total'.

11.5.1 Request URL

/mediafile/count [GET]

11.5.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
owner_id	string	eigenaar van de mediafiles	GET	optioneel
group_id	string	group van de mediafiles	GET	optioneel

11.5.3 Voorbeeld request URL

/mediafile/count?owner_id=t.gebruiker&group_id=groep_a

11.5.4 Response fields

Field	Description
item_count_total	het totaal aantal mediafiles van deze zoekopdracht

12 Media Management overig

12.1 Overhevelen eigenaarschap

Via deze aanroep kunnen de assets, collecties en mediafiles van een bepaalde groep, gebruiker of applicatie naar een nieuwe groep, gebruiker of applicatie worden toegekend.

Voor app_id kan alleen maar de ontvangende applicatie worden opgegeven, let op: hiermee wordt alles definitief verhuisd naar een andere applicatie. Deze actie kan niet meer door de eigenaar ongedaan worden gemaakt.

Bij owner en group moet minimaal 1 oude en 1 nieuwe waarde opgegeven worden.

12.1.1 Request URL

/change_ownership [POST]

12.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
old_owner_id	string	oude eigenaar	POST	optioneel
new_owner_id	string	nieuwe eigenaar	POST	optioneel
old_group_id	alphanumeriek	oude groepsnaam	POST	optioneel
new_group_id	alphanumeriek	nieuwe groepsnaam	POST	optioneel

12.1.3 Voorbeeld request URL

/change_ownership

POST:

old_owner_id=entree004

new_owner_id=entree005

12.1.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

13 User / group management

VP-Core biedt geen volledige set aan user en group management functies. Aangezien er van EGA wordt verwacht dat authenticatie van users in de ega zelf gebeurt, zal daar ook usergegevens opgeslagen moeten worden. De onderstaande user en groupfuncties zijn om gegevens op te halen uit VP-Core van gebruik van users en groepen.

Users en groepen hoeven niet ‘aangemaakt’ te worden binnen VPX. Zodra een asset / collection / mediafile een owner of group wordt toegewezen, kan dit een nog niet ‘bekende’ user of group zijn binnen VPX. In dit soort gevallen heeft deze nieuwe user / group geen quota limiet en valt direct onder het limiet van de EGA.

Bij het opvragen van meerdere users (of groups) worden ‘bekende’ en ‘onbekende’ users (of groups) door elkaar als resultaat gegeven. Het enige verschil bij de ‘onbekende’ users is dat er geen quota staat aangegeven. Een ‘onbekende’ user kun je ‘bekend’ maken door hem een quota toe te wijzen. Hetzelfde geldt voor groups.

Bij het opvragen van een lijst users / groups worden maar beperkt gegevens verbonden aan de users / groups in de lijst. Alleen opvragen van een enkele user of een group kun je de details zien van de user / group.

13.1 Gebruikers opvragen

Via deze call is door een EGA de lijst van gebruikers opvragen. Deze lijst bestaat uit alle gebruikers die een mediafile, asset of collectie bezitten en/of aangemaakt zijn met de user/create REST-call.

13.1.1 Request URL

/user [GET]

13.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	het maximale aantal users dat moet worden weergegeven	GET	verplicht
group_id	string	naam van de groep waar de gebruikers lid van moeten zijn	GET	optioneel
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggegeven.	GET	optioneel

13.1.3 Voorbeeld request URL

/user?limit=10&group_id=test_group_1

13.1.4 Voorbeeld response

```
<items>
  <item id="1">
    <user_id>av-user-zfontys</user_id>
    <group_id>fontys</group_id>
    <quotum>0</quotum>
```

```
<created>2008-04-07 09:53:55</created>
<changed>2008-04-07 09:53:55</changed>
</item>
<item id="2">
  <user_id>admin-beeldengeluid</user_id>
  <group_id>beeldengeluid</group_id>
  <quotum>0</quotum>
  <created>2008-04-07 09:53:55</created>
  <changed>2008-04-07 09:53:55</changed>
</item>
</items>
```

13.1.5 Response fields

Field	Description
user_id	naam van de gebruiker
group_id	naam van de group waar de user eventueel onder valt
quotum	maximaal te gebruiken schijfruimte van de gebruiker
created	Datum en tijd wanneer de gebruiker aangemaakt is (ISO 8601)
changed	Datum en tijd wanneer de gebruiker voor het laatst gewijzigd is (ISO 8601)

13.2 Groepen opvragen

Via deze call is door een EGA de lijst van groepen opvragen.

13.2.1 Request URL

/group [GET]

13.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	het maximale aantal users dat moet worden weergegeven	GET	verplicht
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggeven.	GET	optioneel

13.2.3 Voorbeeld request URL

/group&limit=10

13.2.4 Voorbeeld response

```
<items>
  <item id="1">
    <group_id>fontys</group_id>
    <quotum>0</quotum>
```

```

<created>2008-04-07 09:53:55</created>
<changed>2008-04-07 09:53:55</changed>
</item>
<item id="2">
  <group_id>beeldengeluid</group_id>
  <quotum>0</quotum>
  <created>2008-04-07 09:53:55</created>
  <changed>2008-04-07 09:53:55</changed>
</item>
</items>

```

13.2.5 Response fields

Field	Description
group_id	naam van de groep
quotum	hoeveelheid beschikbare ruimte voor deze groep (0 is ongelimiteerd)
created	datum creatiedatum
changed	datum laatste wijziging.

13.3 Details van een groep opvragen

Via deze call is door een EGA de details van een bepaalde groep op te vragen.

13.3.1 Request URL

/group/{group_id} [GET]

13.3.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
group_id	alfanumeriek	groep naam	URL	verplicht

13.3.3 Voorbeeld request URL

/group/mijngroep

13.3.4 Voorbeeld response

```

<items>
  <item id="1">
    <app_quota_mb>4096000</app_quota_mb>
    <group_quota_mb>0</group_quota_mb>
    <app_diskspace_used_mb>0</app_diskspace_used_mb>
    <group_diskspace_used_mb>126517</group_diskspace_used_mb>
    <quota_available_mb>4096000</quota_available_mb>
    <group_over_quota>false</group_over_quota>
    <created>2008-04-07 09:53:55</created>
    <changed>2008-04-07 09:53:55</changed>
  </item>

```

</items>

13.3.5 Response fields

Field	Description
app_quota_mb	Het totale quotum van de applicatie (megabyte)
group_quota_mb	Het totale quotum van de groep (megabyte)
app_diskspace_used_mb	Het verbruikte quotum van de applicatie (megabyte)
group_diskspace_used_mb	Het verbruikte quotum van de groep (megabyte)
quota_available_mb	Het beschikbare quotum van de groep (megabyte)
group_over_quota	Of de groep over haar quotum zit ('true' of 'false')
created	Datum en tijd wanneer de groep aangemaakt is (ISO 8601)
changed	Datum en tijd wanneer de groep voor het laatst gewijzigd is (ISO 8601)

13.4 Details van een gebruiker opvragen

Via deze call is door een EGA de details van een bepaalde gebruiker op te vragen.

13.4.1 Request URL

/user/{user_id} [GET]

13.4.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user_id	alfanumeriek	naam van de gebruiker	GET	verplicht

13.4.3 Voorbeeld request URL

/user/t.test_user

13.4.4 Response fields

Field	Description
app_quota_mb	Het totale quotum van de applicatie (megabyte)
group_quota_mb	Het totale quotum van de groep (megabyte)
user_quota_mb	Het totale quotum van de gebruiker (megabyte)
app_diskspace_used_mb	Het verbruikte quotum van de applicatie (megabyte)
group_diskspace_used_mb	Het verbruikte quotum van de groep (megabyte)

user_diskspace_used_mb	Het verbruikte quotum van de gebruiker (megabyte)
quota_available_mb	Het beschikbare quotum van de gebruiker (megabyte)
group_over_quota	Of de groep over haar quotum zit ('true' of 'false')
created	Datum en tijd wanneer de gebruiker aangemaakt is (ISO 8601)
changed	Datum en tijd wanneer de gebruiker voor het laatst gewijzigd is (ISO 8601)

13.5 Gebruiker aanmaken

Via deze call is door een EGA een nieuwe gebruiker aan te maken.

13.5.1 Request URL

/user/create [POST]

13.5.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user	alfanumeriek	gebruikersnaam	POST	verplicht
group_id	alfanumeriek	naam van de groep waar de gebruiker in geplaatst dient te worden	POST	optioneel
quotum	numeriek	quotum van de gebruiker in bytes	POST	optioneel (default=0)

13.5.3 Voorbeeld request URL

```
/user/create
user=entree009
quotum=500000
```

13.5.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

13.6 Groep aanmaken

Via deze call is door een EGA een nieuwe groep aan te maken.

13.6.1 Request URL

/group/create [POST]

13.6.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
group_id	alfanumeriek	groepsnaam	POST	verplicht
quotum	numeriek	quotum van de groep in bytes	POST	optioneel (default=0)

Voorbeeld request URL

/group/create

POST:

group_id=mijngroep

13.6.3 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

13.7 Gebruiker verwijderen

Via deze call is door een EGA een gebruiker te verwijderen. Hiermee worden niet de collections/assets/mediafiles van de gebruiker verwijderd. Wel de quota settings van deze user. Dit betekent dat de user na deze call nog steeds zal voorkomen in de /user [GET] lijst. Als de user echt niet meer terug mag komen, moeten eerst van alle objecten de eigenaarschap worden veranderd.

13.7.1 Request URL

/user/{username}/delete [POST]

13.7.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user	alfanumeriek	gebruikersnaam	POST	verplicht

13.7.3 Voorbeeld request URL

/user/alice/delete

13.7.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

13.8 Groep verwijderen

Via deze call is door een EGA een groep te verwijderen, de groep mag geen groepsleden meer hebben.

13.8.1 Request URL

/group/{group_id}/delete [POST]

13.8.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
group_id	alfanumeriek	groepsnaam	URL	verplicht

13.8.3 Voorbeeld request URL

/group/mijngroep/delete

13.8.4 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

13.9 Gebruiker aanpassen

Via deze call is door een EGA een gebruiker aan een groep toe te voegen en zijn quotum te zetten.

13.9.1 Request URL

/user/{user} [POST]

13.9.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
user	alfanumeriek	gebruikersnaam	URL	verplicht
group_id	alfanumeriek	naam van de groep waar de gebruiker in geplaatst dient te worden	POST	verplicht
quotum	numeriek	quotum van de gebruiker	POST	verplicht

13.9.3 Voorbeeld request URL

/user/alice

POST:

group_id=mijngroep

quotum=50000

13.9.4 Response fields

Field	Description
n/a	Bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

13.10 Groep aanpassen

Via deze call is door een EGA een het quotum van een groep aan te passen

13.10.1 Request URL

/group/{group_id} [POST]

13.10.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
group_id	alfanumeriek	groepsnaam	URL	verplicht
quotum	numeriek	quotum van de gebruiker	POST	verplicht

Voorbeeld request URL

/group/mijngroep

POST:

quotum=50000

13.10.3 Response fields

Field	Description
n/a	bij deze call wordt geen data terug gegeven naast de standaard header met de request informatie

14 Statistieken

14.1 Overzicht van de nieuwste mediafiles

Via deze oproep kan een overzicht worden opgevraagd van de 50 nieuwste geuploadede mediafiles.

14.1.1 Request URL

/statistics/newestmediafiles [GET]

14.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	het maximaal aantal resultaten dat teruggegeven dient te worden.	GET	optioneel
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggegeven.	GET	optioneel
order_by	string	Sorteer de lijst op een veld bv: order_by=created Mogelijke sorteer velden: created, filename, owner_id, filesize, container_type	GET	optioneel, default 'created'
order_direction	string	Oplopend ('asc') of aflopend ('desc') sorteren. Voorbeeld: order_direction=desc	GET	optioneel, default 'desc'

Voorbeeld request URL

/statistics/newestmediafiles

14.1.3 Response fields

Field	Description
mediafile_id	De mediafile_id van het bestand.
asset_id	Asset_id waaronder de mediafile valt.
owner_id	Owner id van het bestand.
group_id	Group id van het bestand.
filename	De naam van het bestand.
container_type	Type video container voor zover deze geanalyseerd is.
filesize	Grootte van het bestand voor zover deze geanalyseerd is.
created	Datum waarop de mediafile is aangemaakt.

14.2 Overzicht van de populairste mediafiles

Via deze oproep kan een overzicht worden opgevraagd van de 50 populairste mediafiles die nu in het systeem beschikbaar zijn.

14.2.1 Request URL

/statistics/popularmediafiles [GET]

14.2.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	het maximaal aantal resultaten dat teruggegeven dient te worden.	GET	optioneel
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggegeven.	GET	optioneel
order_by	string	Sorteer de lijst op een veld bv: order_by=created Mogelijke sorteer velden: count, created, filename, owner_id, group_id, filesize, container_type	GET	optioneel, default 'count'
order_direction	string	Oplopend ('asc') of aflopend ('desc') sorteren. Voorbeeld: order_direction=desc	GET	optioneel, default 'desc'

Voorbeeld request URL

/statistics/popularmediafiles

14.2.3 Response fields

Field	Description
mediafile_id	De mediafile_id van het bestand.
asset_id	Asset_id waaronder de mediafile valt.
owner_id	Owner id van het bestand.
group_id	Group id van het bestand.
container_type	Type video container voor zover deze geanalyseerd is.
filesize	Grootte van het bestand voor zover deze geanalyseerd is.
filename	De naam van het bestand.
created	Datum waarop de mediafile is aangemaakt.

count	Het aantal keren dat een ticket voor de mediafile is aangevraagd.
-------	---

14.3 Overzicht van de nieuwste collecties

Via deze oproep kan een overzicht worden opgevraagd van de 50 nieuwste collecties.

14.3.1 Request URL

/statistics/newestcollections [GET]

14.3.2 Request parameters

Parameter	Value	Description	Required/Optional
-----------	-------	-------------	-------------------

Voorbeeld request URL

/statistics/newestcollections

14.3.3 Response fields

Field	Description
coll_id	De collection_id
owner_id	De gebruiker die eigenaar is.
title	De titel behorende bij de collectie.
description	De omschrijving behorende bij de collectie.
created	Het moment waarop de collectie is aangemaakt.

14.4 Overzicht van de populairste collecties

Via deze oproep kan een overzicht worden opgevraagd van de 50 populairste collecties die nu in het systeem beschikbaar zijn.

14.4.1 Request URL

/statistics/popularcollections [GET]

14.4.2 Request parameters

Parameter	Value	Description	Required/Optional
-----------	-------	-------------	-------------------

Voorbeeld request URL

/statistics/popularcollecties

14.4.3 Response fields

Field	Description
coll_id	De collection_id
owner_id	De gebruiker die eigenaar is.
title	De titel behorende bij de collectie.
description	De omschrijving behorende bij de collectie.
created	Het moment waarop de collectie is aangemaakt.
count	Het aantal keren dat een ticket voor de mediafile is aangevraagd.

14.5 Overzicht van de upload van mediafile per maand

Er kan een overzicht worden opgevraagd van de upload van mediafiles per maand (historisch en cumulatief) per gebruikersgroep, eindgebruiker of eindgebruikersapplicatie voor een maximum van 36 maanden.

De gegevens worden per maand getoond. Er kan per maand gebladerd worden in de resultaten.

14.5.1 Request URL

/statistics/dataupload [GET]

14.5.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
year	numeriek	jaar van de rapportage	GET	verplicht

month	numeriek	maand van de rapportage	GET	verplicht
user_id	string	user_id kan gebruikt worden om alleen de resultaten voor 1 user op te vragen	GET	optioneel
group_id	string	group_id kan gebruikt worden om alleen de resultaten voor 1 group op te vragen.	GET	optioneel
limit	numeriek	het maximaal aantal resultaten dat teruggegeven dient te worden.	GET	optioneel
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggegeven.	GET	optioneel

Voorbeeld request URL

/statistics/dataupload

POST:

year=2008

month=2

offset=100

limit=50

14.5.3 Response fields

Field	Description
owner_id	ID van de eigenaar van de geuploadede mediafile
group_id	ID van de groep waar de eigenaar lid van is
file_size	bestandsgrootte van de geuploadede mediafile
timestamp	tijdstip van het uploaden

14.6 Overzicht van schijfruimte per container

Er kan een overzicht worden opgevraagd van de schijfruimte per videobestandstype (container) voor een gebruikersgroep, eindgebruiker of ega. De gegevens worden per maand getoond. Er kan per maand gebladerd worden in de resultaten.

14.6.1 Request URL

/statistics/datausagevideo [GET]

14.6.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
year	numeriek	jaar van de rapportage	GET	verplicht
month	numeriek	maand van de rapportage (1-12)	GET	verplicht

type	string	type kan gebruikt worden om te filteren. Kies uit: container, group of user	GET	verplicht
limit	numeriek	het maximaal aantal resultaten dat teruggegeven dient te worden. (maximaal 200)	GET	verplicht
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggegeven.	GET	optioneel

Voorbeeld request URL

/statistics/datausagevideo?year=2008&month=2&limit=50

14.6.3 Response fields

Field	Description
container_type	het container type van een mediafile die in VP-Core opgeslagen is
user_id of group_id	het onderwerp waar de gebruikte diskruimte betrekking op heeft (niet aanwezig bij type = container)
diskspace_mb	de gebruikte diskruimte in megabytes

14.7 Overzicht van de gevraagde streams

Er kan een overzicht worden opgevraagd van de aangevraagde streams. De informatie dient opgesplitst te worden in gebruikersgroep, eindgebruiker of ega. Er kan ook onderscheid gemaakt worden op type afspeelverzoek (on-demand, live, scheduled en download).

De gegevens worden per maand getoond. Er kan per maand gebladerd worden in de resultaten.

14.7.1 Request URL

/statistics/playedstreams [GET]

14.7.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
year	numeriek	jaar van de rapportage	GET	verplicht
month	numeriek	maand van de rapportage	GET	verplicht
play_type	string	play_type kan gebruikt worden om te filteren. Kies uit: plain, object, download, metafile	GET	optioneel
group_id	string	Specificeer group_id met een naam van een groep om alleen van deze groep resultaat te	GET	optioneel

		zien.		
owner_id	string	Specificeer user_id met een naam van een user om alleen van de user resultaat te zien.	GET	optioneel
limit	numeriek	het maximaal aantal resultaten dat teruggegeven dient te worden.	GET	optioneel
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggeven.	GET	optioneel

Voorbeeld request URL

/statistics/playedstream?year=2008&month=2&offset=0&limit=50

14.7.3 Response fields

Field	Description
mediafile_id	de ID van de afgespeelde mediafile
asset_id	het ID van de asset waar de afgespeelde mediafile toe behoort
owner_id	het ID van de eigenaar van de afgespeelde mediafile
group_id	het ID van de groep waar de eigenaar van de afgespeelde mediafile lid van is
filesize	de bestandsgrootte van de afgespeelde mediafile
container_type	container type van de afgespeelde mediafile
play_type	soort opvraag van de mediafile
played	afspeeldatum

14.8 Overzicht van zoekacties

Er kan een overzicht worden opgevraagd van alle zoekacties. De gegevens worden per maand getoond. Er kan per maand gebladerd worden in de resultaten.

14.8.1 Request URL

/statistics/searchrequest [GET]

14.8.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
year	numeriek	jaar van de rapportage	GET	verplicht
month	numeriek	maand van de rapportage	GET	verplicht
limit	numeriek	het maximaal aantal	GET	optioneel

		resultaten dat teruggegeven dient te worden.		
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggegeven.	GET	optioneel

Voorbeeld request URL

/statistics/searchrequest?year=2008&month=2&offset=0&limit=50

14.8.3 Response fields

Field	Description
n/a	nog in te vullen

14.9 Overzicht van de populairste streams per periode

Er kan een overzicht worden opgevraagd van de meest opgevraagde streams per periode (opgegeven met een startdatum, einddatum). De informatie kan opgesplitst zijn per gebruikersgroep, eindgebruiker, ega en collectie.

De gegevens worden per maand getoond. Er kan per maand gebladerd worden in de resultaten.

14.9.1 Request URL

/statistics/streams [GET]

14.9.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
begindate	string	begindatum van de periode	GET	verplicht
enddate	string	einddatum van de periode	GET	verplicht
owner_id	string	Specificeer een user naam om alleen van deze user resultaat te krijgen.	GET	optioneel
group_id	string	Specificeer een group naam om alleen van deze group resultaat te krijgen.	GET	optioneel
limit	numeriek	het maximaal aantal resultaten dat teruggegeven dient te	GET	optioneel

		worden.		
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggegeven.	GET	optioneel

Voorbeeld request URL

/statistics/streams?startdate=2008-01-01&enddate=2008-02-01&offset=0&limit=50

14.9.3 Response fields

Field	Description
mediafile_id	de ID van de afgespeelde mediafile
requested	het aantal keren dat de mediafile is opgevraagd

14.10 Overzicht van de meest gezochte woorden voor een periode

Er kan een overzicht worden opgevraagd van de 50 meest gezochte woorden voor een bepaalde periode (opgegeven met een startdatum, einddatum).

14.10.1 Request URL

/statistics/searchedwords [GET]

14.10.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
startdate	string	begindatum van de periode	GET	verplicht
enddate	string	einddatum van de periode	GET	optioneel

Voorbeeld request URL

/statistics/searchedwords?startdate=2008-01-01&enddate=2008-02-01

14.10.3 Response fields

Field	Description
word	het gezochte woord
count	aantal keren dat het woord in de periode is gezocht

15 Overige rest calls

Dit hoofdstuk behandelt de rest calls die niet onder de bovenstaande hoofdstukken vallen.

15.1 Opvragen lijst van error codes

Via deze rest call is het mogelijk om een lijst op te vragen van alle error codes die gebruikt worden in VPX.

15.1.1 Request URL

/errorcodes [GET]

15.1.2 Request parameters

Parameter	Waarde	Omschrijving	Methode	Verplicht/Optioneel
limit	numeriek	het maximaal aantal resultaten dat teruggegeven dient te worden.	GET	optioneel
offset	numeriek	het aantal resultaten dat wordt overgeslagen voordat resultaten worden teruggegeven.	GET	optioneel
order_by	string	Sorteer de lijst op een veld bv: order_by=error_id Mogelijke sorteer velden: version, name, code	GET	optioneel, default 'code'
order_direction	string	Oplopend ('asc') of aflopend ('desc') sorteren. Voorbeeld: order_direction=desc	GET	optioneel, default 'desc'

Voorbeeld request URL

/errorcodes?limit=200&offset=10

15.1.3 Response fields

```
<?xml version="1.0"?>
<response>
  <header>
    <item_count>137</item_count>
    <request_process_time>0.0181</request_process_time>
    <request_query_count>8</request_query_count>
    <request_result>success</request_result>
```

```
<request_result_description></request_result_description>

<request_result_id>601</request_result_id>
<request_uri>[GET] errorcodes?
title=warta&title_match=contains&hide_empty_assets=true&limit=10&offset=0&order_by=title&orde
r_direction=asc&operator=and&app_id=5</request_uri>
<vpx_version>1.3.0</vpx_version>

</header>
<items>
<item id="1">
<version>0</version>
<name>HTTP_OK</name>
<code>200</code>
<message></message>

</item>
<item id="2">
<version>0</version>
<name>HTTP_CREATED</name>
<code>201</code>
<message></message>
</item>

.....

</items>
</response>
```

16 CQL - Contextual Query Language

CQL is een standaard dat wordt gebruikt om via een simpele ‘string’ syntax te zoeken op verschillende platformen/databases. CQL staat voor *Contextual Query Language*. Voor VPX betekent dit een manier voor geavanceerd zoeken. Een extra parameter zal worden gebruikt om een CQL string door te geven naar VPX. Deze CQL string wordt binnen VPX omgezet naar een SQL query, die op zijn beurt wordt uitgevoerd op de database. Deze extra parameter is excuusief tenopzichte van de andere zoekparameters. Dat betekent als de CQL string is gedefinieerd, andere zoekparameters niet gebruikt worden en een foutmelding als resultaat geven.

16.1 Basis van CQL

Een CQL string bestaat uit 2 onderdelen die beide optioneel zijn;

[Zoek instructies] [sortby index]

De zoekinstructies zijn 1 of meerdere zoek parameters gevolgd door de sorteer methode. Een zoek instructie bestaat uit 3 onderdelen of 1 onderdeel;

- [index] [relatie] [zoek parameter]
title == “test”
- [zoek parameter]
“test”

De index is het veld naam waarin gezocht moet worden. De relatie geeft de actie aan en de zoek parameter is wat naar gezocht moet worden.

De zoekinstructies worden verbonden door booleans, AND, OR and NOT. De ‘PROX’ boolean wordt niet ondersteund in VPX.

Voorbeeld;

title == “test” OR description == “test”

De ‘sortby’ gedeelte in de CQL bepaald de volgorde van de resultaten. VPX ondersteunt maar 1 dimensie sortering, op 1 index;

sortby title/sort.descending

16.2 Context set

De VPX context set beschrijft voor CQL wat wel en niet kan binnen CQL. Zo is er een vpx context set voor assets en collections. VPX laat alleen zijn eigen vpx context sets toe en zal dus context set definities binnen de CQL alleen van VPX accepteren.

16.3 Relaties

Relaties zijn de definities tussen index en de term.

16.3.1 =

De standaard relatie, gelijk aan. De server zal de bij behorende relatie zoeken voor het type. Bv bij datums zal de relatie == worden gebruikt, ipv =.

Voorbeelden:

1. title = "foo"
(Zoek naar het woord "foo" in title)
2. title = "foo test"
(Zoek naar 1 van deze woorden in title)

16.3.2 ==

Deze relatie geeft aan dat de vergelijking exact moet zijn.

Voorbeelden:

3. title == "foo"
(Zoek naar het woord "foo" in title)
4. title == "foo test"
(Zoek naar de tekst "foo test" in title)

16.3.3 <>

Deze relatie betekent ongelijk aan.

Voorbeelden:

1. title <> "foo"
(title mag niet gelijk zijn aan "foo")

16.3.4 <, >, <=, >=

Deze relaties vergelijken de verhouding tot elkaar (kleiner dan, groter dan, kleiner of gelijk aan, groter of gelijk aan).

Voorbeelden:

1. dc.date > "2008-01-20 00:20:00"

16.3.5 adj

De 'adj' relatie bepaald dat er gezocht wordt binnen de index. Deze relatie is hetzelfde als '=' relatie.

Voorbeelden:

1. dc.title adj "lord of the rings"
(zoek binnen de index naar de tekst "lord of the rings")
2. dc.title adj "blue shirt"
(zoek binnen de index naar "blue", gevolgd door "shirts")

16.3.6 all, any

De 'all' en 'any' relaties worden gebruikt bij zoeken op multi terms. De 'any' relatie is hetzelfde als de '=' relatie.

Voorbeelden:

1. dc.title all "lord rings"
(de index moet de woorden "lord" en "rings" bevaten)
2. dc.title any "lord rings"
(de index moet 1 van de woorden "lord" of "rings" bevaten)

16.3.7 within

De 'within' relatie wordt gebruikt om binnen een range te zoeken. Zo kan dit bij nummers, maar ook bij datums.

1. dc.date within "2005 2006"
(geef alles waarbij dc.date in 2005 en 2006 is)
2. beest.poten within "3 5"
(geef alles waarbij beest.poten tussen 3, 4 en 5 poten heeft)

16.4 Relation modifiers

De relatie aanvulling zijn mogelijke extra parameters die worden meegegeven aan een relatie.

16.4.1 ignoreCase, respectCase

Standaard in de VPX context is ignoreCase; het niet rekening houden met het verschil van normaal of hoofdletters.

1. title =/ignoreCase "foo"
(Zoek naar het woord "foo" in title ongeacht normale of hoofdletters)
2. title =/respectCase "foo"
(Zoek naar het woord "foo" in title maar normale en hoofdletters moeten ook overeen komen)
3. title ==/respectCase "foo"
(Zoek exact naar het woord "foo" in title maar normale en hoofdletters moeten ook overeen komen)

16.5 Masking, wildcards

Masking zijn speciale karakters die gebruikt worden in zoek termen.

16.5.1 Wildcard *

Met dit teken kun je zoeken op meerdere combinaties.

1. dc.title = "l*d"
(geeft alles terug dat begint met een 'l' en eindigt met een 'd')

16.5.2 Masking ?

Met dit teken kun je een character maskeren.

1. `dc.title = "fo?"`
(geef alles terug dat begint met 'fo' en eindigt met elk andere letter)

16.5.3 Anker ^

Met dit teken kun geef je aan dat je iets aan het begin of aan het einde matched. Zie in de voorbeelden het duidelijke verschil tussen

1. `dc.title = "cat^"`
(geef alles terug dat eindigt met 'cat')
2. `dc.title = "^cat"`
(geef alles terug dat begint met 'cat')
3. `dc.title == "^kat in de boom"`
(geef alles terug dat begint met 'kat in de boom')
4. `dc.title == "kat in de boom^"`
(geef alles terug dat eindigt met 'kat in de boom')
5. `dc.title = "^kat in de boom"`
(geef alles terug dat begint met 'kat' of bevat 1 van de woorden 'in de boom')
6. `dc.title = "^kat ^hond boom^"`
(geef alles terug dat begint met 'kat' of 'hond', of eindigt met 'boom')

16.5.4 Backslash

Backslash (\) wordt gebruikt om de tekens '*', '?', aanhalingstekens ("), '^' en zichzelf (\) te escaperen. Een backslash met elk ander teken resulteert in een foutmelding.

1. `dc.title = "\"foo dog\"", enzo"`
(Escape de aanhalingstekens)

16.6 Booleans

De VPX context gebruikt geen gedefinieerde booleans, omdat deze standaard in CQL zitten. Hier dus een lijst van de standaard CQL booleans.

16.6.1 AND

The combinatie van 2 elementen die waar moeten zijn.

16.6.2 OR

The combinatie van 2 elementen waarvan 1 of meer waar moet zijn.

16.6.3 NOT

The combinatie van 2 elementen waarvan links wel waar en de rechter niet waar.

1. `dc.title = foo NOT dc.title = dog`
(title moet woord 'foo' bevatten, maar niet het woord 'dog')

2. NOT dc.title = dog
(*title mag niet het woord 'dog' bevaten*)
3. NOT (dc.title = dog AND dc.title = foo)
(*title mag niet het woord 'dog' en 'foo' bevaten, is hetzelfde als (dc.title != dog AND dc.title != foo)*)

16.6.4 PROX

De boolean 'prox' wordt niet ondersteund door VPX CQL en zal een foutmelding geven.

16.7 Boolean modifiers

Boolean modifiers werken alleen op de 'PROX' boolean en worden daarom niet gebruikt door VPX CQL.

16.8 Asset zoek en sorteer indexen

De VPX index zijn de index namen die gebruikt worden binnen de CQL string om aan te geven waarop te zoeken, bv. 'vpx:dc.creator == "jan"'. De 'vpx:dc' is context naam en is optioneel, 'creator == "jan"' zal ook op dezelfde manier werken. En ook 'dc.creator == "jan"', zal dezelfde werking hebben.

16.8.1 VPX indexes (CQL alias vpx)

- asset_id
- group_id
- owner_id
- provider_id
- reference_id
- mediafile_duration
- mediafile_container_type
- videotimestamp
- videotimestampmodified
- changed
- mime_type
- filename
- numofviews
(de aantal keren dat een asset is bekeken)
- numofplays
(de aantal keren dat een asset afgespeeld is)

16.8.2 Dublin Core (CQL alias vpx:dc / dc)

- title
- creator

- subject
- description
- publisher
- contributor
- date
- type
- format
- identifier
- source
- language
- relation
- coverage_temporal
- coverage_spatial
- rights

16.8.3 *Qualified Dublin Core (CQL alias vpx:qdc / qdc)*

- title_alternative
- description_abstract
- created
- issued
- hasformat
- isformatof
- format_medium
- format_extent
- license
- rightsholder
- isreferencedby

16.8.4 *Content Zoek Profiel (CQL alias vpx:czp / czp)*

- schemanaam
- identificatiecode
- titel
- taal
- beschrijving
- sleutelwoord

- aggregatieniveau
- versie
- status
- soort_leermateriaal
- beoogde_eindgebruiker
- context
- beoogde_leeftijdsgroep
- kosten
- auteursrechten
- omschrijving_rechten
- classificatie

16.9 Collection zoek en sorteer indexen (CQL alias vpx:collection)

- title
- description
- owner_id
- group_id
- created
- changed
- private
- public
- category
- numofvideos

16.10 ACL regels zoeken (CQL alias vpx:acl)

- aut_app
- aut_group
- aut_user
- aut_domain
- aut_realm
- aut_realm_prefix

- `is_protected`

De `acl` context set is special voor het ophalen van bepaalde ACL regels op assets. Het ophalen gebeurt altijd binnen de `app(s)` van de `ega`. Je kunt niet via deze weg asset(s) ophalen waar je als EGA geen toegang tot heeft. Zorg ervoor dat je deze speciale context set alleen gebruikt voor interne doeleinden; zodra je deze context gebruikt zoek je altijd geheel binnen alle assets waar je als EGA/APP toegang tot hebt. Dit betekent dat er geen toegangs checks worden uitgevoerd, behalve de master/slave toegang. Dus je zoekt altijd binnen je eigen assets plus mogelijk slave assets.

Om te zoeken binnen assets die onder een bepaalde `app` vallen (zodat je bv slaves of master kan uitsluiten);

```
aut_app == 5
```

Om alleen assets te tonen die niet van een bepaalde `app` zijn (zodat je bv slaves kan zien, in het geval dat master `app` id 5 heeft);

```
aut_app <> 5
```

Toon mij alle niet beveiligde assets;

```
is_protected == false
```

Toon mij alle beveiligde assets;

```
is_protected == true
```

Toon mij alle assets die via een bepaalde ACL groep gezet zijn;

```
aut_group == '^usergroep 1^'
```

Toon mij alle assets die niet via een bepaalde ACL groep gezet zijn;

```
aut_group <> '^usergroep 1^'
```

Toon mij alle assets die via een bepaalde user is gezet (dit is niet owner);

```
aut_user == '^user^'
```

Toon mij alle assets die niet via een bepaalde user is gezet (dit is niet owner);

```
aut_user <> '^user^'
```

Toon mij alle assets die op een bepaalde domain is gezet;

```
aut_domain == '^foo.nl^'
```

Toon mij alle assets die niet op een bepaalde domain is gezet;

```
aut_domain <> '^foo.nl^'
```

Toon mij alle assets die op een bepaalde realm is gezet;

```
aut_realm == '^@foo.nl^'
```

Toon mij alle assets die niet op een bepaalde realm is gezet;

```
aut_realm <> '^@foo.nl^'
```

Toon mij alle assets die op een bepaalde realm in combinatie met prefix is gezet;

aut_realm == '^@foo.nl^' AND aut_realm_prefix == '^johndoe^'

Toon mij alle assets die niet op een bepaalde realm in combinatie met prefix is gezet;

aut_realm <> '^@foo.nl^' AND aut_realm_prefix <> '^johndoe^'

16.11 Zoek parameters die buiten CQL vallen

Veel parameters zoals 'granted' en 'hide_empty_assets' kunnen niet via CQL worden gezet. Deze moeten dus op de normale manier worden mee gegeven naast de 'cql' parameter.

17 FAQ

Q: Hoe maak ik een lijst met stills (storyboard) van een video?

A: VP-Core slaat per asset 1 still op, dus de EGA zal de stills zelf moeten downloaden en bewaren. De still aanroep accepteert een tijdparameter dat wordt gebruikt om de still te genereren. Deze aanroep kan dus meerdere keren worden aangeroepen, waarna de still kan worden opgehaald en opgeslagen.

Q: Hoe bewaar ik een fragment van een videobestand?

A: VP-Core accepteert start- en eindtijden van assets. Hiermee kan (bv) een zelfgemaakt asx bestand worden geupload waarin een verwijzing staat naar het originele bestand incl start en eindtijd. Dit fragment kan dan afzonderlijk opgevraagd worden zonder dat het fragment dubbel opgeslagen hoeft te worden.

Q: Hoe verwijder ik de objecten van een gebruiker?

A: Er zijn twee mogelijkheden, VP-Core heeft een samengestelde rest call `/change_ownership` (zie beschrijving) ingebouwd waarmee object van eigenaar kunnen worden gewijzigd. Het verwijderen van object van een user is ook mogelijk door eerst de object van die gebruiker op te vragen (parameter: `owner_id=...`) en de desbetreffende `/delete` call aan te roepen.

Q: ROC Rotterdam heeft videos op VPX die alleen afspeelbaar zijn voor die instelling. ROC Den Haag mag ze ook bekijken.

A: Voeg met behulp van `/mediafile/{mediafile_id}/acl` de regels voor de andere instelling toe.

Appendix A – Authenticatie voorbeeld code

```

<?php
/**
 * Copyright (c) 2008 Madcap BV (http://www.madcap.nl)
 * All rights reserved.
 *
 * Permission is granted for use, copying, modification, distribution,
 * and distribution of modified versions of this work as long as the
 * above copyright notice is included
 */

class vpx_connector {
    private $host;
    private $cookie = false;

    public function __construct($host) {
        $this->host = $host;
    }

    public function login($username, $password) {
        $response = $this->request(
            'POST', // method
            '/login', // uri
            array('dbus' => "AUTH DBUS_COOKIE_SHA1 ". $username) // data
        );
        preg_match("@<dbus>DATA vpx 0 (.*)</dbus>@", $response, $matches);
        $challenge = $matches[1];
        $random = substr(md5(microtime(true)), 0, 10);

        $response = $this->request(
            'POST', // method
            '/login', // uri
            array("dbus" => sprintf('DATA %s %s', $random, sha1(sprintf("%s:%s:%s", $challenge, $random, $password)))) // data
        );
        preg_match("@<dbus>(.)</dbus>@", $response, $matches);
        $result = $matches[1];
        return substr($result, 0, 2) === "OK";
    }

    public function request($method, $uri, $data = array()) {
        // build the HTTP header
        switch ($method) {
            case "POST":
                $body = http_build_query($data);
                $header = array(
                    "POST ". $uri ." HTTP/1.0",
                    "Host: ". $this->host,
                    "Content-Length: ". strlen($body),
                    "Content-Type: application/x-www-form-urlencoded",
                );
                break;
            default:

```

```

$header = array(
  "GET ". $uri ." HTTP/1.0",
  "Host: ". $this->host,
);
}
// add the cookie to the request (if set)
if ($this->cookie !== false) {
  $header[] = "Cookie: ". $this->cookie;
}
$request = implode("\r\n", $header) . "\r\n\r\n". $body;
// send the request to VP-Core
$sock = fsockopen($this->host, 80);
 fputs($sock, $request);
// read the response from VP-Core
while ($line = @fgets($sock)) {
  if ($this->cookie === false && strpos($line, "Set-Cookie") === 0) { // collect the cookie (if not set)
    $this->cookie = substr($line, strpos($line, " ") + 1, strpos($line, ";") - strpos($line, " ") - 1);
  }
  $result .= $line;
}
fclose($sock);
list($header, $body) = explode("\r\n\r\n", $result);
// return VP-Core's response
return trim($body);
}
}

$vpv = new vpv_connector("app.acceptatie.vpv.kennisnet.nl"); // hostname without protocol: "hostname"

if ($vpv->login("myapp", "98237498y634t92t5") === TRUE) { // "username", "password"
  echo "Login successful!\n\n";
  // VPX GET example
  echo htmlspecialchars($vpv->request('GET', "/video/transcode_profiles"));
}
else {
  echo "Login failed.\n";
}
?>

```

Appendix B – VPX error codes

Code	Naam	Omschrijving
200	HTTP_OK	
201	HTTP_CREATED	
204	HTTP_NO_CONTENT	
401	HTTP_UNAUTHORIZED	
403	HTTP_FORBIDDEN	
404	HTTP_NOT_FOUND	Page not found
601	ERRORCODE_OKAY	
602	ERRORCODE_TIME_RESTRICTION_START	PP - time_restriction_start error: @date (@timestamp)
603	ERRORCODE_TIME_RESTRICTION_END	PP - time_restriction_end error: @date (@timestamp)
604	ERRORCODE_FILE_NOT_FOUND	PP - @filename doesn't exist!
605	ERRORCODE_DIR_NOT_FOUND	PP - ticket store location doesn't exist! (@location)
606	ERRORCODE_DIR_NOT_WRITABLE	PP - unable to create ticket in @location
607	ERRORCODE_STREAMING_PROFILE_NOT_FOUND	PP - There was no streaming server profile found for the video container used in this mediafile. (video container: '@video_container')
608	ERRORCODE_NO_MEDIAFILE_FOUND_FOR_PROFILE_ID	PP - For the given asset and profile_id no mediafile was found.
609	ERRORCODE_QUERY_ERROR	Incorrect database query
610	STREAM_DOWNLOAD_NOT_ALLOWED	Download not allowed for external streams
611	ERRORCODE_NO_METAFILE_AVAILABLE	This container type doesn't support metafiles (@container_type)
612	ERRORCODE_JOBS_COULD_NOT_BE_STOPPED	Some jobs could not be stopped. Please try again later.
613	ERRORCODE_REST_CALL_IS_DISABLED	Rest call to '@uri' has been disabled, please check documentation.
614	ERRORCODE_UNABLE_TO_CREATE_SYMLINK	Unable to create symbolic link ticket to file.
700	ERRORCODE_ASSET_NOT_FOUND	MM - asset_id=@asset_id was not found in the database
701	ERRORCODE_MEDIAFILE_NOT_FOUND	MM - mediafile_id=@mediafile_id was not found in the database
702	ERRORCODE_INVALID_APP_ID	MM - invalid application id (app_id)

703	ERRORCODE_INVALID_USER_ID	MM - the specified user_id (@user_id) is not the owner of this asset
704	ERRORCODE_UNKNOWN_ERROR_SAVING_METADATA	MM - unknowns error while saving metadata for asset_id @asset_id
705	ERRORCODE_EMPTY_RESULT	MM - The query returned an empty result set (i.e. zero rows).
706	ERRORCODE_COLLECTION_NOT_FOUND	MM - coll_id=@coll_id was not found in the database
707	ERRORCODE_COLLECTION_ASSET_RELATION_ALREADY_EXISTS	MM - the asset is already present in the collection
708	ERRORCODE_COLLECTION_ASSET_RELATION_NOT_FOUND	MM - the asset/collection relation was not found in the database
709	ERRORCODE_ASSET_NOT_EMPTY	MM - there are still @mediafile_count mediafile(s) present in this asset.
710	ERRORCODE_COLLECTION_NOT_EMPTY	MM - there are still @asset_count asset(s) present in this collection.
711	ERRORCODE_ASSET_SUPPLEMENT_NOT_FOUND	MM - Supplement @supplement_id not found.
712	ERRORCODE_WEBSERVICE_DISABLED	The webservice has been disabled
713	ERRORCODE_UNKNOWN_VPX_VERSION	Unknown VPX version requested: @version
714	ERRORCODE_INVALID_ASSET_MEDIAFILE_COMBINATION	MM - invalid asset/mediafile combination
715	ERRORCODE_NO_CHANGES	Insufficient parameters given, changes not saved
716	ERRORCODE_INVALID_FAV_TYPE	Invalid favorite type (@type), valid favorite types; (@valid_types)
717	ERRORCODE_INVALID_TIMESTAMP	@name is not a valid datetime or timestamp value, must be in format of YYYY-MM-DD HH:MM:DD / YYYY-MM-DD HH:MM / YYYY-MM-DD HH / YYYY-MM-DD / YYYY-MM / YYYY
718	ERRORCODE_UNEXPECTED_ERROR	A unexpected program error / problem happened
719	ERRORCODE_MISSING_TIMESTAMP	@name must have 2 dates separated with a
720	ERRORCODE_CANT_PLAY_MEDIAFILE	This mediafile (@mediafile_id) can not be played; @reason.
721	ERRORCODE_CANT_TRANSCODE_MEDIAFILE	This mediafile (@mediafile_id) can not be transcoded; @reason.
722	ERRORCODE_APP_DISABLED	This application has been disabled
800	ERRORCODE_UNKNOWN_JOB	Jobs - unknown job specified

1000	ERRORCODE_VALIDATE_INVALID_TYPE	VPX - invalid type specified: @type
1001	ERRORCODE_VALIDATE_INVALID_LENGTH_USE	VPX - 'length' can not be used in combination with 'length_{min max}'
1002	ERRORCODE_VALIDATE_REQUIRED_PARAMETER	VPX - missing required parameter: @param (@type)
1003	ERRORCODE_VALIDATE_FAILED	VPX - parameter validation failed: @param (@type)
1004	ERRORCODE_SORT_FIELD_ERROR	VPX - sorting on '@field' is not allowed
1005	ERRORCODE_SORT_DIRECTION_ERROR	VPX - invalid sort direction specified: '@sort_direction'. Use 'ASC' or 'DESC' instead.
1006	ERRORCODE_PP_INVALID_TIME	PP - parameter validation failed: @param (@type) should be less than 87 400 000 msec (=24h)
1007	ERRORCODE_VALIDATE_INVALID_LANGUAGE	VPX - 'language' parameter validation failed. Please use the 2-letter codes as defined in ISO 639-1.
1008	ERRORCODE_INVALID_SUPPLEMENT	Invalid XML or maximum allowed size exceeded
1009	ERRORCODE_MIX_OF_URI_AND_FILE	A single mediafile can't have a URI and a filename!
1010	ERRORCODE_CHANGE_URI_AND_FILE	A URI-based mediafile can't be changed to a file-based mediafile and visa vers.
1011	ERRORCODE_CHANGE_OWNERSHIP_MISSING_PARAMETERS	No 'old' or 'new' value present
1012	ERRORCODE_INSUFFICIENT_PARAMETERS	Insufficient parameters given
1013	ERRORCODE_SUPPLEMENT_LIMIT_REACHED	Maximum number of supplements for this asse reached (@max)
1014	ERRORCODE_MEDIAFILE_DOWNLOAD_DISABLED	Download is disabled for this mediafile
1015	ERRORCODE_UPLOAD_TARGET_IS_NOT_AN_ORIGINAL_FILE	Uploading to a non-original mediafile is not allowed
1016	ERRORCODE_QUOTA_REACHED	Quota limit reached. Allowed size: @quota MB. Used size: @used MB.
1017	ERRORCODE_HOSTNAME_ALREADY_IN_GROUP	The hostname (@hostname) is already a member of the autorisation group (@group).
1018	ERRORCODE_HOSTNAME_NOT_FOUND	The hostname (@hostname) was not found in the autorisation group (@group).
1019	ERRORCODE_GROUP_NOT_EMPTY	There are still one or more hostnames present in this autorisation group (@group)
1020	ERRORCODE_VALIDATE_SEARCH_SYNTAX_FAILED	Search parameters syntax contains a syntax error, please check your search_* parameters
1021	ERRORCODE_VALIDATE_SEARCH_SYNTAX_FAILED_UNEXPECTED_END	Search parameters syntax ends abruptly

1022	ERRORCODE_VALIDATE_SEARCH_SYNTAX_FAILED_IN_VALID_FIELD	Search parameters: you can't search on field '@field'
1023	ERRORCODE_METADATA_DEFINITION_NOT_EMPTY	There are still one or more properties present in this property definition
1024	ERRORCODE_INVALID_METADATA_DEFINITION	Invalid metadata definition
1025	ERRORCODE_HOSTNAME_TYPE_NO_MATCH_GROUP_TYPE	The hostname (@hostname) with type (@hostname_type) must be the same type as the group (@group) with type (@group_type).
1026	ERRORCODE_AUT_GROUP_ALREADY_EXISTS	The authorization group (@group) already exists.
1027	ERRORCODE_AUT_GROUP_NOT_FOUND	The authorization group (@group) was not found.
1028	ERRORCODE_AUT_NAME_NOT_FOUND	The authorization hostname (@hostname) was not found.
1029	ERRORCODE_INVALID_DATA_PARAMETER	VPX - @parameter contains invalid data, expecting @data
1030	ERRORCODE_FTP_BATCH_NOT_FOUND	The FTP batch id (@batch_id) was not found.
1031	ERRORCODE_CANT_RETYPE_GROUP_MUST_BE_EMPTY	You can't change the type of group '@group', when it contains hostnames.
1032	ERRORCODE_CQL_EXCLUSIVE	VPX - When you define a CQL search string, you can not specify any other search or order by parameter.
1033	ERRORCODE_CQL_ERROR	VPX - CQL returned an error; @error.
1034	ERRORCODE_METADATA_DEFINITION_ALREADY_EXISTS	VPX - Metadata definition already exists.
1100	ERRORCODE_RETRIEVING_JOBLIST	VPX - error retrieving joblist from the database
1101	ERRORCODE_RETRIEVING_JOBSTATUS	VPX - error retrieving jobstatus from the database
1102	ERRORCODE_WRITING_JOBSTATUS	VPX - error writing jobstatus to the database
1103	ERRORCODE_WRITING_JOBPROGRESS	VPX - error writing jobprogress to the database
1104	ERRORCODE_WRITING_UNKNOWN_JOBSTATUS	VPX - error writing jobstatus to the database, an invalid status code was sent.
1105	ERRORCODE_RETRIEVING_TRANSCODELIST	VPX - error retrieving transcodelist from the database.
1106	ERRORCODE_CREATING_JOB	VPX - error creating a new job
1107	ERRORCODE_DELETING_JOB	VPX - error deleting a job
1108	ERRORCODE_UNKNOWN_JOB_TYPE	VPX - unknown job type

1109	ERRORCODE_RETRIEVING_TRANSCODE_PROFILE	VPX - error retrieving transcode profile from the database.
1110	ERRORCODE_RETRIEVING_DEFAULT_TRANSCODE_PROFILE	VPX - error retrieving default transcode profile.
1111	ERRORCODE_NO_DEFAULT_TRANSCODE_PROFILE	VPX - no default transcode profile specified.
1112	ERRORCODE_UNKNOWN_TRANSCODE_PROFILE	VPX - unknown transcode profile.
1113	ERRORCODE_NO_TRANSCODE_PARAMETERS	VPX - no transcoding parameters provided.
1114	ERRORCODE_JOB_TRANSCODE_PARAMETER_NOT_FOUND	VPX - transcoding parameter @key is unknown.
1115	ERRORCODE_JOB_TRANSCODE_PARAMETER_TOO_LOW	VPX - transcoding parameter @key has a value of @value, which is lower then the minimum @min_value.
1116	ERRORCODE_JOB_TRANSCODE_PARAMETER_TOO_HIGH	VPX - transcoding parameter @key has a value of @value, which is higher then the maximum @max_value.
1117	ERRORCODE_JOB_TRANSCODE_PARAMETER_WRONG_VALUE	VPX - transcoding parameter @key has an invalid value of @value.
1118	ERRORCODE_JOB_TRANSCODE_PARAMETER_NOT_FLOAT	VPX - transcoding parameter @key has an invalid value of @value.
1119	ERRORCODE_JOB_TRANSCODE_PARAMETER_COMBINATION	VPX - transcoding parameter combination of tool @tool and file extension @file_extension does not exist.
1120	ERRORCODE_JOB_TRANSCODE_TIMEOUT	VPX - error, transcode job has exceeded the max transcode job runtime (@timeout s)
1121	ERRORCODE_CREATING_TRANSCODE_JOB	VPX - error creating a transcode job.
1122	ERRORCODE_JOB_NOT_FOUND	VPX - job_id=@job_id was not found in the database
1123	ERRORCODE_JOB_ASSET_NOT_FOUND	VPX - asset_id=@asset_id was not found in the database
1124	ERRORCODE_CREATING_UPLOAD_JOB	VPX - error creating a upload job
1125	ERRORCODE_CREATING_ANALYSE_JOB	VPX - error creating an analyse job
1126	ERRORCODE_JOB_MEDIAFILE_NOT_FOUND	VPX - mediafile_id=@mediafile_id was not found in the database
1127	ERRORCODE_JOB_USER_NOT_FOUND	VPX - user_id=@user_id was not found in the database
1128	ERRORCODE_JOB_COULD_NOT_BE_REMOVED	VPX - Job kon niet worden verwijderd (job_id=@job_id)

1129	ERRORCODE_JOB_FRAMETIME_GREATER_THEN_DURATION	chosen frametime is higher dan duration of the mediafile.
1130	ERRORCODE_RETRIEVING_ASSET	VPX - error retrieving the asset from database
1131	ERRORCODE_JOB_CANCEL_NOT_ALLOWED	VPX - deleting of job not allowed
1132	ERRORCODE_JOB_CREATE_NOT_ALLOWED	VPX - creation of job is not allowed
1140	ERRORCODE_UPLOAD_ALREADY_EXISTS	VPX - a upload is already active for this asset_id
1150	ERRORCODE_STILL_NOT_FOUND	MM - could not find still for asset_id=@asset_id.
1201	ERRORCODE_RETRIEVING_SERVERLIST	JH - error retrieving serverlist
1202	ERRORCODE_RETRIEVING_AVAILABLE_SERVERLIST	JH - error retrieving server-job-list
1301	ERRORCODE_STARTING_JOB_FAILED	Jobserver - starting of job failed
1302	ERRORCODE_RETRIEVING_SERVER_JOBLIST	Jobserver - error retrieving jobs on server
1401	ERRORCODE_STUB_SERVER_RETRIEVING_JOB_FOR_EXECUTION	Jobserver - error retrieving 1 job for execution
1500	ERRORCODE_INVALID_UPLOAD_TICKET	Upload - invalid upload ticket
1501	ERRORCODE_CREATE_MEDIAFILE_DURING_UPLOAD	Upload - couldn't create mediafile during upload
1502	ERRORCODE_CANNOT_COPY_MEDIAFILE	Upload - couldn't copy mediafile to final destination
1503	ERRORCODE_NOT_ENOUGH_FREE_QUOTA	Upload failed because of too less free quota.
1600	ERRORCODE_DBUS_PROTOCOL_ERROR	Authenticate - identification process
1601	ERRORCODE_ACCESS_DENIED	Authenticate - access denied
1602	ERRORCODE_ACCESS_DENIED_INVALID_APP_ID	Authenticate - access denied (invalid app_id)
1700	ERRORCODE_FTP_ERROR	FTP users - error
1701	ERRORCODE_FTP_CREDENTIAL_LENGTH	FTP users - credentials not of minium length
1702	ERRORCODE_FTP_UNKNOWN_USER	FTP users - unknown user
1703	ERRORCODE_FTP_USER_EXISTS	User already exists
1800	ERRORCODE_NOT_AUTHORIZED	Action is not allowed, not authorized (@reason)
1801	ERRORCODE_ACL_ERROR_AUTHORIZED	
1900	ERRORCODE_USERMAN_INVALID_GROUP	Invalid group name
1901	ERRORCODE_USERMAN_GROUP_NOT_EMPTY	The group is not empty
1902	ERRORCODE_USERMAN_INVALID_USER	Invalid username
1903	ERRORCODE_USERMAN_GROUP_EXISTS	Group already exists

1904	ERRORCODE_USERMAN_USER_EXISTS	User already exists
------	-------------------------------	---------------------