



ecoDMS API

HANDBUCH

Impressum

ecoDMS Build 21.02
Handbuch-Datum: 15. Juni 2021
Typ: ecoDMS API
Sprache: DE

Ersteller / Urheber: ecoDMS GmbH
©2021 Dresdener Straße 1
52068 Aachen
Deutschland
Webseite: www.ecodms.de
E-Mail: info@ecodms.de
Telefon: 0049 241 47572 01
Sitz der Gesellschaft: Aachen
Registergericht: Amtsgericht Aachen 19201
Geschäftsführung: Dipl.-Betw. Michael Schmitz
Helge Lühmann

Wichtige Informationen

Dieses von der Firma ecoDMS GmbH bereitgestellte Benutzerhandbuch ist urheberrechtlich geschützt. Jegliche Verwendung von Texten und Abbildungen, insbesondere des Nachdrucks, die der Übersetzung oder des Einstellens in elektronische Systeme im Ganzen oder in Auszügen bedarf der vorherigen, schriftlichen Genehmigung durch die Firma ecoDMS GmbH, Dresdener Straße 1, 52068 Aachen. Auf die etwaige Strafbarkeit urheberrechtswidrigen Handelns wird hingewiesen.

KEINE GARANTIE. Alle technischen Angaben und Programmdarstellungen in diesem Benutzerhandbuch wurden mit größter Sorgfalt erarbeitet. Trotzdem sind Fehler nicht ganz auszuschließen. Die Firma ecoDMS GmbH sieht sich daher gezwungen, darauf hinzuweisen, dass weder eine Garantie, noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für Verbesserungsvorschläge und Hinweise auf Fehler ist die Autorin dankbar.

Inhaltsverzeichnis

1	Einleitung	6
2	API Dienst Aktivierung	7
2.1	Dienst starten	7
2.2	Dienst stoppen	7
3	Statistik	8
3.1	Einstellungen in ecoDMS	8
3.2	Abruf via ecoDMS API	8
3.2.1	Beispiel Ausgabe "apiStatistics"	8
4	Swagger Online-Dokumentation	9
4.1	Starten	9
4.2	Stoppen	9
5	Anmeldung & Autorisierung im ecoDMS API REST Service	10
5.1	Verfügbare Funktionen: REST-Commands	10
5.1.1	Verbindung testen	10
5.1.2	Anmeldung und Autorisierung	10
5.2	Parameter	10
5.2.1	archived ID (required)	11
5.2.2	ecoDMSUser (required)	11
5.2.3	ecoDMSPassword (required)	11
5.2.4	Beispiel: Anmeldung und Authentifizierung	11
5.2.5	Verbindung beenden	11
6	Suchfilter	12
6.1	Parameter	12
6.1.1	searchFilterList (required)	13
6.1.2	Beispiel-Werte	13
6.2	Returns	13
6.2.1	docInfoList	13
6.2.2	Beispiel JSON-Liste	13
6.3	Verfeinerte Suchfunktionen	13
7	Barcodes	15
7.1	Beispiel Ausgabe "readBarcodes"	15
8	Aufruf der ecoDMS-Attribute	16
8.1	Status	16
8.1.1	Returns	16
8.1.1.1	statuslist	16
8.1.1.2	Beispiel JSON-Liste	16
8.2	Rollen / Berechtigungen	16
8.2.1	Liste aller Rollen	16
8.2.2	Liste der Benutzer-Rollen	17
8.2.3	Returns	17
8.2.3.1	Übersicht aller Rollen	17
8.2.3.2	Beispiel Übersicht (alle Rollen)	17
8.2.3.3	Beispiel Übersicht (Benutzer-Rollen)	17

8.3	Dokumentenarten	17
8.3.1	Returns	17
8.3.1.1	docTypeList	17
8.3.1.2	Beispiel Übersicht	18
8.4	Ordner	18
8.4.1	Returns	18
8.4.1.1	folderList	18
8.4.1.2	Beispiel Übersicht	18
9	Dokumente und Versionen	19
9.1	Parameter	19
9.1.1	docID (required)	19
9.1.2	versionNr (optional)	19
9.1.3	Beispiel REST-Command	20
9.2	Returns	20
9.2.1	documentObject	20
10	Archivierung / Versionierung neuer Dokumente	21
10.1	Versionsdetails abrufen	21
10.1.1	Beispiel Ausgabe "readDocumentVersions"	22
10.2	Datei nach ecoDMS übertragen	22
10.2.1	Parameter	22
10.2.1.1	file (required)	22
10.2.1.2	versionControlled (required)	23
10.2.2	Returns	23
10.2.2.1	docID	23
10.3	Datei und PDF ins ecoDMS übertragen	23
10.3.1	Parameter	23
10.3.1.1	file (required)	23
10.3.1.2	pdfFile (required)	23
10.3.1.3	versionControlled (required)	23
10.4	Returns	23
10.4.1	docID	23
10.5	Neue Version hinzufügen	24
10.5.1	Parameter	24
10.5.1.1	file (required)	24
10.5.1.2	docID (required)	24
10.5.1.3	fixed / finalised (required)	24
10.5.2	Returns	24
10.5.2.1	docID	24
10.6	Neue Version mit PDF hinzufügen	24
10.6.1	Parameter	25
10.6.1.1	file (required)	25
10.6.1.2	docID (required)	25
10.6.1.3	fixed (required)	25
10.6.2	Returns	25
10.6.2.1	docID	25
11	Klassifizierung und Dokumenten-Informationen	26
11.1	Dokumenten-Informationen ausgeben	26
11.1.1	Parameter	26
11.1.1.1	docID (required)	26
11.1.2	Returns	26
11.1.2.1	docInfoObject	26
11.2	Klassifizierung durchführen	27
11.2.1	Parameter	27
11.2.1.1	docID (required)	27
11.2.1.2	docInfoObject (required)	27

11.2.2 Returns	27
11.2.2.1 docID	27
11.3 Liste definierter Klassifizierungsattribute ausgeben	28
11.3.1 Returns	28
11.3.1.1 classificAttrList	28
11.3.1.2 Beispiel Übersicht	28
11.4 Liste der Standard-Klassifizierungsattribute	28
11.5 Mehrfachklassifizierung anlegen	29
11.5.1 Parameter	29
11.5.1.1 docInfoObject (required)	29
11.5.2 Returns	29
11.5.2.1 cldocID	29
12 Neuen Ordner erstellen	30
12.1 Parameter	30
12.1.1 dmsOrdnerObject (required)	30
12.1.2 parentoid (optional)	30
12.2 Returns	30
12.2.1 ordnerID	30
12.3 Ordner auf oberster Ebene anlegen	31
13 Dokumentenvorschau	32
13.1 Parameter	32
13.1.1 docID (required)	32
13.1.2 pageNr (required)	32
13.1.3 height (required)	32
13.2 Returns	32
13.2.1 Datei-Stream	32
13.2.2 Vorschau-Größe	32
14 Verlinkungen	33
14.1 Verlinken	33
14.2 Anzeigen	33
14.3 Löschen	33
15 Löschfunktion	34
15.1 In Papierkorb verschieben	34
15.2 Wiederherstellen	34
16 Upload-Größe	35
16.1 Windows	35

1 Einleitung

- Zur Verwendung der ecoDMS API sind s.g. API-Connects erforderlich. Diese können optional zur bestehenden ecoDMS Lizenznummer erworben werden
- Die Kosten für die ecoDMS API sind abhängig von der Anzahl monatlich benötigter API-Connects (Staffelpreise)
- 1 API-Connect = Monatlich 1 Upload oder Download über die ecoDMS API
- Das genaue Lizenzmodell ist auf der ecoDMS Webseite zu finden: <https://www.ecodms.de/index.php/de/ecodms-api/lizenzmodell>

Speziell für Entwickler gibt es diese Schnittstelle. Die ecoDMS API ermöglicht den Anschluss von beliebigen Drittsystemen wie z.B. CRM-Software, Warenwirtschaftssysteme uvm. Über die Schnittstelle sind die grundlegenden Funktionen des ecoDMS Servers wie zum Beispiel „archivieren“, „klassifizieren“ oder „herunterladen“ ansprechbar. Der Aufruf der einzelnen Funktionen erfolgt über REST Web Services. So hat jede Funktion eine eindeutige Adresse, die als URL ausgedrückt wird und unter anderem in Internetbrowsern verwendet werden kann.

Sollte z.B. ein Dokument mit der ID=5 aus ecoDMS heruntergeladen werden, wird der Befehl

```
'/document/5'
```

verwendet, wobei *'/document'* den REST-Aufruf der jeweiligen Funktion darstellt und *'5'* der Eingabeparameter ist, welches als Dokument ID=5 interpretiert wird.

Bitte beachten Sie: ecoDMS API REST Service unterstützt nur die UTF-8-Kodierung. D.h vor jedem Aufruf jeder REST-Funktion muss unbedingt sichergestellt werden, dass der jeweilige HTTP-Request in UTF-8 kodiert ist.

1. Base URL:

```
http://hostname:[port]/api/[REST-Command]
```

2 API Dienst Aktivierung

Um die ecoDMS API einsetzen zu können, muss der Dienst zunächst über den Einstellungsdialog im ecoDMS Client aktiviert werden.

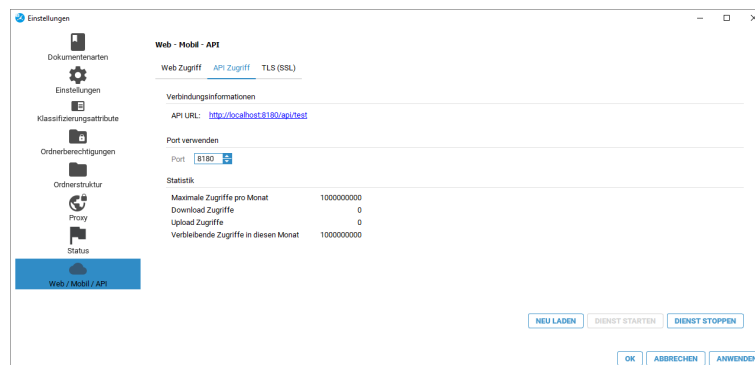


Abb. (ähnlich) 2.1: Einstellungen: API Zugriff (API REST Service)

2.1 Dienst starten

Um den API-Dienst zu starten, gehen Sie wie folgt vor:

1. Öffnen Sie als Administrator die Einstellungen im ecoDMS Client
2. Klicken Sie auf den Reiter "Web / Mobil / API -> API Zugriff "
3. **Port verwenden:** Standardmäßig ist der Port "8180" eingestellt. Diesen können Sie optional ändern.
4. **Dienst starten:** Klicken Sie auf "Dienst starten" um den API Zugriff mit den getätigten Einstellungen zu starten
 - a) Der API Zugriff wird aktiviert. Dieser Vorgang kann ein wenig dauern.
5. **API Url:** Hier zeigt ecoDMS den Link zum Aufrufen der ecoDMS API im eigenen Netzwerk (also intern) an
6. Klicken Sie auf "Anwenden" um den Vorgang zu speichern oder brechen Sie diesen mit "Abbrechen" ab.

2.2 Dienst stoppen

Um den API-Dienst zu stoppen, gehen Sie wie folgt vor:

1. Öffnen Sie als Administrator die Einstellungen im ecoDMS Client
2. Web / Mobil / API -> API Zugriff -> Dienst stoppen
 - a) Der API Zugriff wird gestoppt. Dieser Vorgang kann ein wenig dauern.
3. Verbindungsinformationen -> Webclient Url: Der Link zur ecoDMS API ist nicht verfügbar
4. Klicken Sie auf "Anwenden" um den Vorgang zu speichern oder brechen Sie diesen mit "Abbrechen" ab.

3 Statistik

Die Anzahl der verfügbaren und im laufenden Monat bereits gebrauchten API-Connects kann über die ecoDMS API selbst und im Einstellungsdialog von ecoDMS abgerufen werden.

3.1 Einstellungen in ecoDMS

1. Öffnen Sie als Administrator die Einstellungen im ecoDMS Client
2. Web / Mobil / API -> API Zugriff
3. Die Statistik zeigt folgende Werte zur ecoDMS API an:
 - a) **Maximale Zugriffe pro Monat:** Verfügbare Anzahl an monatlichen API-Connects
 - b) **Download Zugriffe:** Anzahl der bereits abgerufenen bzw. heruntergeladenen Dokumente über die ecoDMS API im aktuellen Monat
 - c) **Upload Zugriffe:** Anzahl der bereits archivierten bzw. hochgeladenen Dokumente über die ecoDMS API im aktuellen Monat
 - d) **Verbleibende Zugriffe in diesem Monat:** Anzahl der verbleibenden API-Connects im aktuellen Monat

3.2 Abruf via ecoDMS API

1. Mit folgendem Befehl kann die Statistik über die ecoDMS API aufgerufen werden:

```
/api/apiStatistics
```

2. Diese Statistik zeigt folgende Werte zur ecoDMS API an:
 - a) **Maximale Zugriffe pro Monat:** Verfügbare Anzahl an monatlichen API-Connects
 - b) **Download Zugriffe:** Anzahl der bereits abgerufenen bzw. heruntergeladenen Dokumente über die ecoDMS API im aktuellen Monat
 - c) **Upload Zugriffe:** Anzahl der bereits archivierten bzw. hochgeladenen Dokumente über die ecoDMS API im aktuellen Monat

3.2.1 Beispiel Ausgabe "apiStatistics"

```
{  
  "uploadCount": "0",  
  "maxCount": "30",  
  "downloadCount": "50"  
}
```


4 Swagger Online-Dokumentation

Die Swagger UI kann zum Testen der Zugriffe und als Dokumentation der API genutzt werden.

4.1 Starten

1. Mit folgendem Befehl kann die Swagger UI gestartet werden:

```
/api/startDebug
```

2. Zum Starten von Swagger wird der API-Dienst im Hintergrund neu gestartet.
 - a) Dies kann einige Sekunden dauern.
3. Es erfolgt eine automatische Weiterleitung auf die Startseite von swagger-ui.
4. Nach dem erfolgreichen Start ist die Seite über den folgenden Link erreichbar:

```
<servername>/swagger-ui.html#/eco-dms-rest-controller
```

Eine ausführliche Anleitung zur Verwendung von Swagger kann auf der Webseite des Projektes aufgerufen werden.

```
https://swagger.io/
```

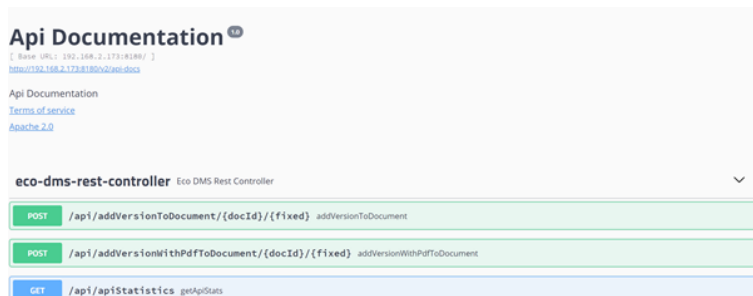


Abb. (ähnlich) 4.1: Beispiel zur API Dokumentation

4.2 Stoppen

Sofern der API-Port produktiv genutzt wird oder für jeden erreichbar ist, ist es sinnvoll Swagger zu stoppen.

1. Die Swagger-Oberfläche wird mit folgendem Aufruf gestoppt.

```
<server-name>/api/stopDebug
```

2. Zum Stoppen von Swagger wird der API-Dienst im Hintergrund neu gestartet.
 - a) Dies kann einige Sekunden dauern.

5 Anmeldung & Autorisierung im ecoDMS API REST Service

Der API REST Service von ecoDMS verwendet die HTTP-Basic-Authentifizierung nach RFC 2617.

1. Der erste Schritt bei der Anmeldung ist die Herstellung der Verbindung mit dem ecoDMS Server. Dies wird mit dem Aufruf der REST-Funktion *'connect'* realisiert. Dabei muss unbedingt ein vorhandenes ecoDMS Archiv ausgewählt werden. Als Eingabeparameter dieser Funktion wird die ID-Nummer des Archivs angegeben.

a) Beispiel:

```
http://localhost:8180/api/connect/1
```

2. Sowie die Verbindung mit einem ecoDMS Archiv erfolgreich hergestellt wurde, müssen die eingegebenen ecoDMS-Benutzerdaten (username und password) auf ihre Gültigkeit geprüft werden. Dies geschieht, indem man das Base64-codierte Authentifizierungsmuster im Header des HTTP-Requests an ecoDMS API REST Service sendet. Dafür kann die REST-Funktion *'/test'* aufgerufen werden (siehe Beschreibung zur Funktion *'/test'*).

a) Beispiel:

```
http://localhost:8180/api/test
```

3. HTTP-Response liefert im Header unter anderem auch die Cookie-Informationen, die vom Client lokal gespeichert und für die weiteren Transaktionen im HTTP-Request-Header übergeben werden sollen.

5.1 Verfügbare Funktionen: REST-Commands

5.1.1 Verbindung testen

```
HttpMethod.GET /test
```

Testet die Verbindung zum ecoDMS API REST Service und benötigt auch keine Anmeldung. Bei erfolgreicher Verbindung erscheint eine Meldung in der Art "aplord Test: OK".

5.1.2 Anmeldung und Autorisierung

```
HttpMethod.GET /connect/{archiveId}
```

Dient der Anmeldung und Autorisierung eines schon bestehenden ecoDMS-Benutzers am ecoDMS API REST Service. Dabei ist die Angabe der ID des entsprechenden ecoDMS Archivs notwendig. Es sind keine Transaktionen möglich, solange sich der Benutzer vom API REST Service nicht mit den gültigen ecoDMS-Benutzerdaten autorisiert hat (s. Bsp. Anmeldung und Autorisierung).

1. Resource-URL:

```
http://host:[port]/api/connect/1
```

5.2 Parameter

Sie können verschiedene Parameter verwenden.

5.2.1 archived ID (required)

ID-Nummer des Archivs, welches für die weiteren Transaktionen verwendet werden soll.

1. Beispiel-Wert:

```
1
```

5.2.2 ecoDMSUser (required)

Es handelt sich hierbei um den ecoDMS-Benutzernamen. Dieser muss zusammen mit dem Benutzer-Passwort als Base64-codiertes Authentifizierungsmuster im HTTP-Header an den Server übermittelt werden.

1. Beispiel-Wert:

```
1
```

5.2.3 ecoDMSPassword (required)

Es handelt sich hierbei um das ecoDMS-Benutzer-Passwort. Dieses muss zusammen mit dem Benutzernamen als Base64-codiertes Authentifizierungsmuster im HTTP-Header an den Server übermittelt werden.

1. Beispiel-Wert:

```
musteruser
```

5.2.4 Beispiel: Anmeldung und Authentifizierung

```
musteruser:mypassword
```

Ausgabe Base64 codiert (HTTP-Basic-Authentifizierung)

```
Basic bXVzdGVydXNlcjpteXBhc3N3b3Jk
```

5.2.5 Verbindung beenden

```
HttpMethod.GET /disconnect
```

Dieser Befehl beendet die mit dem ecoDMS API REST Service aufgebaute Verbindung. Als Antwort wird eine entsprechende Meldung geliefert, die das Abmelden des aktuellen ecoDMS-Benutzers bestätigt.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/disconnect
```

2. Beispiel-Antwort:

```
Logout for user musteruser successful
```

6 Suchfilter

```
HttpMethod.POST /searchDocuments
```

Liefert eine Liste mit ecoDMS-Dokument-Information-Objekten (ecoDocInfo) aus dem gewählten ecoDMS Archiv, die mit Hilfe eines vom Benutzer definierten Suchfilters zusammengestellt wird. Jedes ecoDocInfo-Objekt enthält folgende Attribute:

- Dokumenten-ID
- Klassifizierungs-ID
- ID des ecoDMS Archivs, in welchem sich das jeweilige Dokument befindet
- ecoDMS-Klassifizierungsattribute inkl. der Beschreibungen und Werte (z.B. Hauptordner, Ordner, Bemerkung, Status, Revision, Dokumentenart, Datum usw.)

Der Suchfilter ähnelt der Erweiterten Suche im ecoDMS Client und kann mehrere Suchkriterien beinhalten. Jedes einzelne Suchkriterium besteht aus folgenden 3 Teilen:

1. Klassifizierungsattribut
2. Operator
3. Wert

Beispiel: Hier wird eine Liste von Dokumenten mit dem Status "Wiedervorlage" ausgegeben:

```
'status=2'
```

Darstellung in JSON:

```
[{"classifyAttribut":"status", "searchOperator":"=", "searchValue":"2"}]
```

Zum Definieren des Suchfilters werden folgende REST-Funktionen benötigt:

```
/types  
/status  
/folders
```

Diese Funktionen liefern die für die Suche erforderlichen IDs (Status-IDs, Ordner-IDs, Dokumentenart-IDs...). Bei einer Suchanfrage werden die Werte der entsprechenden Klassifizierungsattribute ausschließlich nach den IDs und nicht nach den Namen gesetzt. Soll beispielsweise der Inhalt eines ecoDMS-Ordners ermittelt werden, muss die exakte Ordner-ID angegeben werden. Beim Suchen nach dem Inhalt eines Ordners mit der ID = 1.4.5.2 -ohne Berücksichtigung seiner Unterordner- muss die vollständige Ordner-ID gesetzt sein und nicht nur der erste Teil.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/searchDocuments
```

6.1 Parameter

Für die Suche können verschiedene Parameter verwendet werden.

6.1.1 searchFilterList (required)

Es handelt sich hierbei um eine Liste mit den einzelnen Suchfiltern. Diese Liste muss im HTTP-Body als JSON-Nachricht an den API REST Service übermittelt werden.

6.1.2 Beispiel-Werte

```
docid > 0; docid <= 10; docart=2; folder=1.3
```

```
[{"classifyAttribut":"docid", "searchOperator":">", "searchValue":"0"},
{"classifyAttribut":"docid", "searchOperator":"<=", "searchValue":"10"},
{"classifyAttribut":"docart", "searchOperator":"=", "searchValue":"2"},
{"classifyAttribut":"folder", "searchOperator":"=", "searchValue":"1.3"}
{"classifyAttribut":"folder", "searchOperator":"ilike", "searchValue":"TEXT"}]
```

– **Information zum Beispiel:** Zur Suche nach einer Textstelle / einem Wort wird der searchOperator "ilike" benötigt.

6.2 Returns

6.2.1 docInfoList

Dies ist eine Liste mit den ecoDMS Dokument-Informations-Objekten als JSON-List.

6.2.2 Beispiel JSON-Liste

```
[{
  {"DocID":15,"clDocId":15,"archiveName":"1","classifyAttributes":
  {"changeid":"ecoDMS","status":"3","mainfolder":"1","timestamp":"2013-05-10
  09:58:39","dyn_000011366794281":"","folder":"1.3","revision":"1.3",
  "docid":"15#15","docart":"1","bemerkung":"Musterdokument","rechte":"W",
  "defdate":"","cdate":"2013-05-10"},"editRoles":null,"readRoles":null}
},{
  {"DocID":23,"clDocId":23,"archiveName":"1","classifyAttributes":
  {"changeid":"ecoDMS","status":"3","mainfolder":"6","timestamp":"2013-06-25
  12:49:34","dyn_000011366794281":"","folder":"6.6","revision":"1.5",
  "docid":"23#23","docart":"21","bemerkung":"Musterdokument","rechte":"W",
  "defdate":"","cdate":"2013-05-10"},"editRoles":null,"readRoles":null}
}]
```

6.3 Verfeinerte Suchfunktionen

```
HttpMethod.POST /searchDocumentsExt
```

Die verfeinerte Suchfunktion funktioniert identisch zur normalen Suche, unterstützt aber zusätzlich noch vier URL-Parameter.

Aufruf Beispiel:

```
/api/searchDocumentsExt?maxDocumentCount=100&personalDocumentsOnly=false&readRoles=false&
trashedDocuments=true
```

maxDocumentCount Anzahl der zurückgelieferten Dokumente (max 1.000)

personalDocumentsOnly=true Ansicht von ausschließlich persönlichen Dokumenten (default=false)

readRoles=false Lesen der Berechtigungen des Dokuments (Performance Tuning) (default = true)

trashedDocuments=true Lesen der Dokumente aus den Papierkorb (default = false)

Der Aufbau der Suchfilter und die Antwortdaten entsprechen dem Vorgehen vom Abschnitt "Parameter" in diesem Kapitel.

7 Barcodes

```
/api/readBarcodes
```

Dieser Befehl liefert Details zu erkannten Barcodes auf einem Dokument. Folgende Werte werden hierbei ausgegeben:

- Anzahl der Barcodes auf einem Dokument
- Seite
- Barcodetyp
- Barcode Daten

7.1 Beispiel Ausgabe "readBarcodes"

```
{
  "barcodeCount": 0,
  "barcodes":
    [
      {
        "barcodeData": "string",
        "page": 0,
        "type": "DMTX"
      }
    ]
}
```

8 Aufruf der ecoDMS-Attribute

Jedes einzelne ecoDMS-Klassifizierungsattribut kann aufgerufen werden.

8.1 Status

```
HttpMethod.GET /status
```

Dieser Befehl liefert eine Liste mit den in ecoDMS definierten Status. Jeder Status hat eine eindeutige ID und Bezeichnung. Der Status "Wiedervorlage" beinhaltet beispielsweise folgende Attribute:

- ID=2
- Name=Wiedervorlage

Die ID wird unter anderem benötigt, um mit Hilfe eines Suchfilters innerhalb des ecoDMS Archivs nach dem entsprechenden Wert des Attributs "Status" zu suchen. Des Weiteren kann über die Status-ID der passende Wert innerhalb einer Klassifizierung zu einem bestimmten Dokument geändert werden.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/status
```

8.1.1 Returns

8.1.1.1 statuslist

Diese Liste beinhaltet die in ecoDMS definierten Status als JSON-List.

8.1.1.2 Beispiel JSON-Liste

```
[{"id":3,"name":"Erledigt"}, {"id":2,"name":"Wiedervorlage"}, {"id":1,"name":"Zu Bearbeiten"}]
```

8.2 Rollen / Berechtigungen

8.2.1 Liste aller Rollen

```
HttpMethod.GET /roles
```

Dieser Befehl liefert eine Liste mit den in ecoDMS hinterlegten Rollen (Benutzer, Gruppen). Über diese Werte kann beispielsweise bei der Klassifizierung eine Berechtigung übergeben werden.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/roles
```


8.2.2 Liste der Benutzer-Rollen

```
Http.Methot GET /userRoles
```

Dieser Befehl listet die ecoDMS-Rollen des aktuell verbunden Benutzers auf.

8.2.3 Returns

8.2.3.1 Übersicht aller Rollen

Dies ist eine Liste mit den in ecoDMS definierten Status als JSON-List.

8.2.3.2 Beispiel Übersicht (alle Rollen)

```
["Buchhaltung","ecoICELogon","ecoSIMSADMIN","ecoSIMSALLDOCS","ecoSIMSCCLASSIFY",  
"ecoSIMSCREATEFOLDER","ecoSIMSHISTORY","ecoSIMSUSER","ecoSIMSVERSIONING","ecoSIMSWEBCLIENT", "  
ecoSIMSTEMPLATES", "ecoSIMSDELETE","Geschäftsführung","r_ecodms","scanner","Vertrieb"]
```

8.2.3.3 Beispiel Übersicht (Benutzer-Rollen)

```
["ecoSIMSADMIN", "ecoSIMSUSER", "ecoSIMSCCLASSIFY", "ecoSIMSHISTORY", "ecoSIMSALLDOCS", "  
ecoSIMSWEBCLIENT", "ecoSIMSTEMPLATES", "ecoSIMSDELETE", "r_j.doe"]
```

8.3 Dokumentenarten

```
HttpMethod.GET /types
```

Liefert eine Liste mit den in ecoDMS definierten "Dokumentenarten". Jede Dokumentenart hat eine eindeutige ID, eine entsprechende Bezeichnung und eine Aufbewahrungsfrist.

Beispiel anhand der Dokumentenart "Rechnung":

- Id: '5'
- Name: 'Rechnung'
- Frist: {"jahre":0,"monate":0,"tage":0}.

Die ID wird unter anderem benötigt, um mit Hilfe eines Suchfilters innerhalb des ecoDMS Archivs nach dem entsprechenden Wert des Attributs Dokumentenart zu suchen. Des Weiteren kann über die Dokumentenart-ID der passende Wert innerhalb einer Klassifizierung zu einem bestimmten Dokument geändert werden.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/types
```

8.3.1 Returns

8.3.1.1 docTypeList

Diese Liste beinhaltet die in ecoDMS definierten Dokumentenarten als JSON-List.

8.3.1.2 Beispiel Übersicht

```
[
{"id":1,"name":"Anfrage","frist":{"jahre":0,"monate":0,"tage":0}},
{"id":2,"name":"Angebot","frist":{"jahre":0,"monate":0,"tage":0}},
{"id":3,"name":"Anschreiben","frist":{"jahre":0,"monate":0,"tage":0}},
{"id":4,"name":"Bescheid","frist":{"jahre":0,"monate":0,"tage":0}},
{"id":5,"name":"Rechnung","frist":{"jahre":0,"monate":0,"tage":0}}
]
```

8.4 Ordner

```
HttpMethod.GET /folders
```

Liefert eine Liste aller Ordner, die in der Ordnerstruktur von ecoDMS angelegt sind. Jedes Ordner-Objekt beinhaltet folgende Informationen:

- Eindeutige Ordner-ID
- Bezeichnung des Ordners
- Schlüssel
- Schlagwörter
- mainfolder-Variable

Als Hauptordner deklarierte Ordner in ecoDMS werden mit dieser Variable versehen.

Die eindeutige Ordner-ID des jeweiligen Ordners wird zum Beispiel bei der Definition eines Suchfilters benötigt.

```
folder=1.3
```

Bei Angabe der gesuchten Ordner-ID wird der Inhalt des angegebenen Verzeichnisses, sowie der Inhalt aller darunter liegender Ordner zurückgegeben. Wenn bei der Klassifizierung die Werte der Klassifizierungsattribute "Hauptordner" oder "Ordner" betroffen sind, werden ebenfalls die Ordner-IDs der jeweiligen Ordner benötigt (nicht die Ordner-Bezeichnungen).

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/folders
```

8.4.1 Returns

8.4.1.1 folderList

Diese Liste beinhaltet die in ecoDMS definierten Verzeichnissen als JSON-List.

8.4.1.2 Beispiel Übersicht

```
[{"mainFolder":false,"oId":"1","foldername":"Kunden","externalKey":"10000","buzzwords":""},
{"mainFolder":false,"oId":"1.1","foldername":"Musterkunde01","externalKey":"","buzzwords":""},
{"mainFolder":false,"oId":"1.3","foldername":"Musterkunde03","externalKey":"","buzzwords":""},
{"mainFolder":false,"oId":"2.1","foldername":"Musterlieferant01","externalKey":"20100","buzzwords":""}]
```

9 Dokumente und Versionen

```
HttpMethod.GET /document/{docID}
```

Diese Funktion lädt das angeforderte Dokument mit der übergebenen "DocID" aus ecoDMS.

```
HttpMethod.GET /document/{docID}/version/{versionNr}
```

Sollte eine bestimmte Version eines Dokuments heruntergeladen werden, wird hierfür ebenfalls die Funktion verwendet. Allerdings muss ergänzend zur DocID die Version angegeben werden. Hierzu wird in der URL der Zusatz *'/version/versionNr'* verwendet.

Die Funktion liefert als Ergebnis ein ecoDMS-File-Objekt, welches das in ecoDMS gespeicherte Dokument repräsentiert. Jedes ecoDMS-File-Objekt besitzt die folgende Struktur:

docid	38
Dateigröße	98890
Dokumentename	38BestellungMusterfirma.pdf
Dateityp	.txt, .pdf, .doc, usw. -> hier: .pdf
Datei-Stream	byte

Die Dateiinformationen sind im HTTP-Body der HTTP-Response enthalten.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/document/38  
http://host:[port]/api/document/38/version/2
```

9.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

9.1.1 docID (required)

DocID steht für "Dokumenten-Identifikationsnummer". Sie wird automatisch und fortlaufend nach der Archivierung für jedes archivierte Dokument vergeben. Die ID-Nummer wird direkt in der URL angegeben. Das zur DocID gehörige Dokument kann dann heruntergeladen werden.

Beispielwert:

```
38
```

9.1.2 versionNr (optional)

Die Versionsverwaltung ermöglicht die Archivierung beliebig vieler Versionen eines Dokuments in ecoDMS. Jede neue Version wird mit einer hochgezählten Versionsnummer als neues Dokument archiviert. Über diesen Befehl kann eine bestimmte

Version abgerufen werden.

Beispielwert:

```
/version/2
```

9.1.3 Beispiel REST-Command

Die zweite Version des in ecoDMS archivierten Dokuments mit der DocID 38 soll heruntergeladen werden.

```
/document/38/version/2
```

9.2 Returns

9.2.1 documentObject

Der Datei-Stream wird im HTTP-Body der HTTP-Response-Nachricht übermittelt. Alle Dateiinformationen können im HTTP-Header der HTTP-Response-Nachricht entnommen werden.

10 Archivierung / Versionierung neuer Dokumente

Zum Archivieren eines neuen Dokuments in ecoDMS können folgende REST-Funktionen verwendet werden:

```
/uploadFile  
/documentInfo  
/classifyDocument
```

1. /uploadFile

Zunächst muss die erforderliche Datei auf ecoDMS Server in das gewünschte ecoDMS Archiv hochgeladen werden. Dazu wird die API REST Service Funktion *'/uploadFile'* verwendet. Als Ergebnis liefert die Funktion die Dokument-ID des im ecoDMS neu archivierten Dokuments.

2. /documentInfo

Sowie das Dokument erfolgreich auf dem ecoDMS Server hochgeladen wurde und eine entsprechende DocID vorliegt, kann die Datei optional klassifiziert werden. Einige Klassifizierungsattribute wie z.B. "Datum", "Letzte Änderung", etc. werden beim Archivieren eines neuen Dokuments vom ecoDMS Server automatisch vergeben. Diese Informationen müssen vor dem Klassifizieren aus ecoDMS heraus ermittelt werden. Diese Funktionalität wird mit dem REST-Befehl *'/documentInfo'* realisiert. Als Ergebnis wird ein ecoDMS-Dokument-Information-Objekt (ecoDocInfo) geliefert. Die Struktur eines *'ecoDocInfo'*-Objekts sieht wie folgt aus:

- Dokumenten-ID
- Klassifizierungs-ID
- ID-Nummer des ecoDMS Archivs, in welchem sich das jeweilige Dokument befindet
- ecoDMS-Klassifizierungsattribute, mit Bezeichnung und entsprechenden Werten (Hauptordner, Ordner, Bemerkung, Status, Revision, Dokumentenart, Datum, etc.)

Die Informationen werden in Form einer JSON-Nachricht vom API REST Service bereitgestellt (siehe REST-Funktion *'/documentInfo'*). Sollte ein Wert eines bestimmten Klassifizierungsattributs geändert werden, muss die Änderung auch an entsprechender Stelle im *'ecoDocInfo'*-Objekt (JSON) vorgenommen werden. Bitte beachten Sie, dass als Wert für ein Klassifizierungsattribut nur die IDs verwendet werden dürfen und nicht deren Bezeichnungen.

3. /classifyDocument

Wenn das *'ecoDocInfo'*-Objekt (JSON) mit allen notwendigen Informationen versehen ist, kann die API REST Service Funktion *'/classifyDocument'* aufgerufen werden. Hierbei muss mindestens der Dateiname angegeben werden. Dieser Eintrag wird als Wert für das Klassifizierungsattribut "Bemerkung" im neuen ecoDMS Dokument interpretiert. Im Kapitel "Klassifizierung und Dokumenten-Informationen" können weitere Details zur Klassifizierung nachgelesen werden.

10.1 Versionsdetails abrufen

```
/api/document/docid/readDocumentVersions
```

Dieser Befehl liefert Details zu den Versionen von einem Dokument. Folgende Werte werden hierbei ausgegeben:

- DocID

- Datum
- Benutzer
- Dateiname
- Abgeschlossen: ja/nein
- Abgeschlossen Benutzer
- Abgeschlossen Datum
- Bemerkung
- PDF Datei

10.1.1 Beispiel Ausgabe "readDocumentVersions"

```
[
  {
    "id": 1,
    "date": "2021-03-05 09:16:01",
    "user": "apiuser",
    "file": "test.pdf",
    "fixed": false,
    "fixUser": "",
    "fixDate": null,
    "comment": "",
    "pdfFile": "test.pdf",
    "checkoutUser": "",
    "checkoutDate": null
  }
]
```

10.2 Datei nach ecoDMS übertragen

```
HttpMethod.POST /uploadFile/{versionControlled}
```

Diese Funktion überträgt eine neue Datei nach ecoDMS und liefert als Ergebnis die Dokument-ID des archivierten Dokuments.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/uploadFile/false
```

10.2.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

10.2.1.1 file (required)

Es handelt sich hierbei um die Datei, die in ecoDMS gespeichert werden soll. Der Datei-Stream wird im HTTP-Body der Request-Nachricht an den API REST Service übermittelt.

10.2.1.2 versionControlled (required)

Dieser Befehl zeigt an, ob es sich um ein versioniertes Dokument handelt oder nicht.

Beispielwerte:

- *false*: Für ein nicht versioniertes Dokument
- *true*: Für ein versioniertes Dokument

10.2.2 Returns

10.2.2.1 docID

Die DocID wird automatisch und fortlaufend nach der Archivierung für jedes archivierte Dokument vergeben. Es handelt sich hierbei um die ID des neu archivierten Dokuments.

10.3 Datei und PDF ins ecoDMS übertragen

```
HttpMethod.POST /uploadFileWithPDF/{versionControlled}
```

Diese Funktion überträgt eine neue Datei und das dazugehörige PDF ins ecoDMS. Das Speichern einer PDF-Datei erfolgt beispielsweise über den ecoDMS PDF/A Drucker oder über ein ecoDMS Office Plug-in. Als Ergebnis wird die Dokument-ID des archivierten Dokuments geliefert.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/uploadFileWithPdf/true
```

10.3.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

10.3.1.1 file (required)

Es handelt sich hierbei um die Datei, die in ecoDMS gespeichert werden soll. Der Datei-Stream wird im HTTP-Body der Request-Nachricht an den API REST Service übermittelt.

10.3.1.2 pdfFile (required)

Dies ist die Datei, die in ecoDMS als Standard-Dokument angezeigt wird (z.B. PDF).

10.3.1.3 versionControlled (required)

Dieser Befehl zeigt an, ob es sich um ein versioniertes Dokument handelt oder nicht. Beispielwerte:

- *false*: Für ein nicht versioniertes Dokument
- *true*: Für ein versioniertes Dokument

10.4 Returns

10.4.1 docID

Die DocID wird automatisch und fortlaufend nach der Archivierung für jedes archivierte Dokument vergeben. Es handelt sich hierbei um die ID des neu archivierten Dokuments.

10.5 Neue Version hinzufügen

```
HttpMethod.POST /addVersionToDocument/{docID}/{fixed}
```

Mit diesem Aufruf kann eine neue Version zu einem bereits in ecoDMS archivierten Dokument hinzugefügt werden. Die neue Datei wird als weitere Version zum angegebenen Dokument in ecoDMS hinterlegt. Dabei ist die Angabe erforderlich, ob die aktuell geladene Version des Dokuments fixiert werden soll oder nicht. Wenn "Dokument abschließen" gesetzt wird, können nach der Archivierung keine weiteren Versionen mehr zu diesem Dokument hinzugefügt werden.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/addVersionToDocument/15/false
```

10.5.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

10.5.1.1 file (required)

Es handelt sich hierbei um die Datei, die in ecoDMS gespeichert werden soll. Der Datei-Stream wird im HTTP-Body der Request-Nachricht an den API REST Service übermittelt.

10.5.1.2 docID (required)

Hier wird docID des Dokuments hinterlegt, zu welchem die Datei als neue Version hinzugefügt werden soll. Beispielwert:

```
15
```

10.5.1.3 fixed / finalised (required)

Wenn "Dokument abschließen" gesetzt wird, können nach der Archivierung keine weiteren Versionen mehr zu diesem Dokument hinzugefügt werden. Beispielwerte:

- *false*: Version nicht fixieren
- *true*: Version fixieren

10.5.2 Returns

10.5.2.1 docID

Die DocID wird automatisch und fortlaufend nach der Archivierung für jedes archivierte Dokument vergeben. Es handelt sich hierbei um die ID des neu archivierten Dokuments.

10.6 Neue Version mit PDF hinzufügen

```
HttpMethod.POST /addVersionWithPdfToDocument/{docID}/{fixed}
```

Mit diesem Aufruf kann eine neue Version inklusive PDF-Datei zu einem bereits in ecoDMS archivierten Dokument hinzugefügt werden. Die neue Datei wird als weitere Version zum angegebenen Dokument in ecoDMS hinterlegt. Dabei ist die Angabe erforderlich, ob die aktuell geladene Version des Dokuments fixiert werden soll oder nicht. Wenn "Dokument abschließen" gesetzt wird, können nach der Archivierung keine weiteren Versionen mehr zu diesem Dokument hinzugefügt werden.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/addVersionWithPdfToDocument/15/false
```

10.6.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

10.6.1.1 file (required)

Es handelt sich hierbei um die Datei, die in ecoDMS gespeichert werden soll. Der Datei-Stream wird im HTTP-Body der Request-Nachricht an den API REST Service übermittelt.

10.6.1.2 docID (required)

Dies ist die Datei, die in ecoDMS als Standard-Dokument angezeigt wird.

- Zum Beispiel : PDF

10.6.1.3 fixed (required)

Wenn "Dokument abschließen" gesetzt wird, können nach der Archivierung keine weiteren Versionen mehr zu diesem Dokument hinzugefügt werden. Beispielwerte:

- *false* :Version nicht fixieren
- *true* : Version fixieren

10.6.2 Returns

10.6.2.1 docID

Die DocID wird automatisch und fortlaufend nach der Archivierung für jedes archivierte Dokument vergeben. Es handelt sich hierbei um die ID des neu archivierten Dokuments.

11 Klassifizierung und Dokumenten-Informationen

Mit Klassifizierung ist die Zuordnung und Vergabe von Dokumenten-Informationen und deren Berechtigung gemeint. Für jede archivierte Datei können der Ordner, die Dokumentenart, der Status, die Berechtigungen und weitere Details vergeben werden. Die Klassifizierungsattribute und Dokumenten-Informationen aus ecoDMS können vergeben, angezeigt und auch verändert werden.

11.1 Dokumenten-Informationen ausgeben

```
HttpMethod.GET /documentInfo
```

Liefert ein ecoDMS-Dokument-Information-Objekt (`ecoDocInfo`) zu dem in ecoDMS archivierten Dokument unter Angabe der entsprechenden Dokument-ID. Dieses Objekt enthält folgende Attribute:

- Dokumenten-ID
- Klassifizierungs-ID
- ID-Nummer des ecoDMS Archivs, in welchem sich das jeweilige Dokument befindet
- ecoDMS-Klassifizierungsattribute, mit deren Bezeichnung und entsprechenden Werten (Hauptordner, Ordner, Bemerkung, Status, Revision, Dokumentenart, Datum, etc.)

Diese Dokumenten-Informationen werden zum Beispiel für die Klassifizierung benötigt. Es können sowohl neue Klassifizierungen vergeben werden, als auch Änderungen an bestehenden Klassifizierungen durchgeführt werden.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/documentInfo/38
```

11.1.1 Parameter

Es gibt folgende Parameter für diese Funktion.

11.1.1.1 docID (required)

Es handelt sich hierbei um die ID-Nummer des archivierten Dokuments, zu welchem die Klassifizierungsattribute und Dokumenten-Informationen benötigt werden. Die ID wird direkt in der URL angegeben.

Beispielwert:

```
38
```

11.1.2 Returns

11.1.2.1 docInfoObject

Das ecoDMS-Dokument-Information-Objekt kann im HTTP-Header der HTTP-Response-Nachricht als JSON-Nachricht entnommen werden.

Beispiel:

JSON-Struktur von einem ecoDMS-Dokument-Information-Objekt (`ecoDocInfo`)

```
{ "DocID":10, "clDocId":10, "archiveName":"1", "classifyAttributes":  
{ "changeid":"ecoDMS", "status":"3", "mainfolder":"1", "ctimestamp":"2013-05-10 09:59:09", "  
  dyn_000011366794281":""," "folder":"1.3", "revision":"1.3", "docid":"10#10", "docart":"19", "  
  bemerkung":"Musterdokument", "rechte":"W", "defdate":""," "cdate":"2013-05-10"}, "editRoles":null  
  , "readRoles":null}
```

11.2 Klassifizierung durchführen

```
HttpMethod.POST /classifyDocument
```

Über diesen Befehl wird das in ecoDMS archivierte Dokument klassifiziert. Dabei werden die aktuellen Werte von allen erforderlichen Klassifizierungsattributen gesetzt. Hierzu muss mit Hilfe der REST-Funktion *'/documentInfo'* das ecoDMS-Dokument-Information-Objekt (ecoDocInfo) zum jeweiligen Dokument abgefragt werden.

Das Dokument-Information-Objekt muss mit den aktuellen Klassifizierungswerten angepasst werden. Im Anschluss kann das Objekt im Body des HTTP-Requests an den API REST Service übermittelt werden.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/classifyDocument
```

11.2.1 Parameter

Es gibt folgende Parameter für diese Funktion.

11.2.1.1 docID (required)

Es handelt sich hierbei um die ID-Nummer des archivierten Dokuments, zu welchem die Klassifizierung erfolgen soll. Die ID wird direkt in der URL angegeben.

Beispielwert:

```
38
```

11.2.1.2 docInfoObject (required)

Es handelt sich hierbei um das ecoDMS-Dokument-Information-Objekt mit den aktuellen Klassifizierungswerten. Beispiel: JSON-Struktur von einem ecoDMS-Dokument-Information-Objekt (ecoDocInfo).

```
{ "DocID":10, "clDocId":10, "archiveName":"1", "classifyAttributes":  
{ "changeid":"ecoDMS", "status":"3", "mainfolder":"1", "ctimestamp":"2013-05-10 09:59:09", "  
  dyn_000011366794281":""," "folder":"1.3", "revision":"1.3", "docid":"10#10", "docart":"19", "  
  bemerkung":"Musterdokument", "rechte":"W", "defdate":""," "cdate":"2013-05-10"}, "editRoles":null  
  , "readRoles":null}
```

11.2.2 Returns

11.2.2.1 docID

Die DocID wird automatisch und fortlaufend nach der Archivierung für jedes archivierte Dokument vergeben. Es handelt sich hierbei um die ID des klassifizierten Dokuments.

11.3 Liste definierter Klassifizierungsattribute ausgeben

```
HttpMethod.GET /classifyAttributes
```

Über diesen Aufruf kann eine Liste mit sämtlichen in ecoDMS definierten Klassifizierungsattributen ausgegeben werden. Jeder Eintrag in der Liste enthält folgende Struktur:

- "Spaltenname in der Datenbank" : "Bezeichnung in ecoDMS"

Diese Funktion kann z.B verwendet werden, wenn die Spaltenbezeichnungen der dynamischen Attribute benötigt werden.

1. Resource-URL:

```
http://host:[port]/api/classifyAttributes
```

11.3.1 Returns

11.3.1.1 `classificAttrList`

Dies ist eine Liste mit den in ecoDMS definierten Klassifizierungsattributen als JSON-List.

11.3.1.2 Beispiel Übersicht

```
{
  "changeid":"Bearbeitet von","status":"Status",
  "mainfolder":"Hauptordner","ctimestamp":"LetzteÄnderung",
  "dyn_000011366794281":"Auftragsnummer","folder":"Ordner",
  "docid":"DocId","revision":"Revision","dyn_000011380876313":"Paul02",
  "docart":"Dokumentenart","bemerkung":"Bemerkung",
  "rechte":"Berechtigung","dyn_000011380867530":"Test2",
  "defdate":"Wiedervorlage ab","cdate":"Datum"
}
```

11.4 Liste der Standard-Klassifizierungsattribute

Bei der Zuordnung / Klassifizierung können verschiedene Attribute vergeben werden. Hierbei wird zwischen Standardattributen und dynamischen Attributen unterschieden. Die Bezeichnungen von dynamischen Klassifizierungsattributen (Attribute, die in ecoDMS vom Administrator manuell definiert wurden), können mit Hilfe der REST-Funktion *'documentInfo'* abgefragt werden.

Beispiel: Bezeichnung eines dynamischen Klassifizierungsattributs:

- dyn_000011366794281

Nachfolgend eine Liste der Standard-Klassifizierungsattribute:

```
ECO_CLASSIFY_MAINFOLDER = "mainfolder";
ECO_CLASSIFY_FOLDER = "folder";
ECO_CLASSIFY_STATUS = "status";
ECO_CLASSIFY_DOCTYPE = "docart";
ECO_CLASSIFY_BEMERKUNG = "bemerkung";
ECO_CLASSIFY_DOCID = "docid";
```

```
ECO_CLASSIFY_REVISION = "revision";  
ECO_CLASSIFY_FULLTEXT = "fulltext";  
ECO_CLASSIFY_RESUBMISSION = "defdate";  
ECO_CLASSIFY_DATE = "cdate";  
ECO_CLASSIFY_MODIFIED_BY = "changeid";  
ECO_CLASSIFY_MODIFIED= "ctimestamp";
```

11.5 Mehrfachklassifizierung anlegen

```
HttpRequest.POST /createNewClassify
```

Anlegen von zusätzlichen Klassifizierungen (Mehrfachklassifizierungen) zu einem Dokument. Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Beispiel-URL:

```
http://host:[port]/api/createNewClassify
```

11.5.1 Parameter

Es gibt folgende Parameter für diese Funktion.

11.5.1.1 docInfoObject (required)

Es handelt sich hierbei um das ecoDMS-Dokument-Information-Objekt mit den aktuellen Klassifizierungswerten. Beispiel: JSON-Struktur von einem ecoDMS-Dokument-Information-Objekt (ecoDocInfo).

```
{"DocID":10,"clDocId":10,"archiveName":"1","classifyAttributes":  
{"changeid":"ecoDMS","status":"3","mainfolder":"1","ctimestamp":"2013-05-10 09:59:09",  
"dyn_000011366794281":"","folder":"1.3","revision":"1.3","docid":"10#10","docart":"19",  
"bemerkung":"Musterdokument","rechte":"W","defdate":"","cdate":"2013-05-10"},"editRoles":null  
,"readRoles":null}
```

11.5.2 Returns

11.5.2.1 cldocID

ecoDMS liefert die eindeutige Klassifizierungs-ID für die angelegte Mehrfachklassifizierung.

12 Neuen Ordner erstellen

```
HttpMethod.POST /createFolder/parent/{parentoid}
```

Mit dieser Funktion wird ein neuer Ordner in der ecoDMS-Ordnerstruktur erstellt.

- Sollte ein neuer Ordner in der obersten Ordner-Ebene angelegt werden, sind die Werte für *'parent'* und *'parentoid'* nicht erforderlich.
- Falls ein Unterordner in einem bereits existierenden ecoDMS Ordner angelegt werden muss, so muss der Zusatz *'/parent/'* gefolgt von der entsprechenden Ordner-ID des übergeordneten Ordner in der URL mit angegeben werden.

Zum Definieren eines neuen Ordners wird das ecoDMS Ordner-Objekt verwendet. Jedes ecoDMS Ordner-Objekt beinhaltet folgende Informationen:

- Eindeutige Ordner-ID
- Bezeichnung des Ordners
- ecoDMS Schlüssel
- Schlagwörter
- mainfolder-Variable

Als Hauptordner deklarierte Ordner in ecoDMS werden mit dieser Variable versehen.

Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:[port]/api/createFolder/parent/2.1
```

12.1 Parameter

Es gibt hierbei folgende Parameter:

12.1.1 dmsOrdnerObject (required)

Das ecoDMS-Ordner-Objekt beinhaltet alle erforderlichen Parameter zum Erstellen eines Ordners.

12.1.2 parentoid (optional)

Es handelt sich hierbei um die "Ordner-ID" des übergeordneten Ordners (Überordner, Vaterordner).
Beispielwert, falls ein Unterordner aus ecoDMS mit der ID=2.1 benötigt wird:

```
2.1
```

12.2 Returns

12.2.1 ordnerID

Die Ordner-ID wird automatisch vom System vergeben. Es handelt sich hierbei um die ID des neu erstellten Ordners.

12.3 Ordner auf oberster Ebene anlegen

```
HttpRequest.POST /createFolder
```

Anlegen eines neuen Ordners ohne übergeordnetem Ordner.
Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Beispiel-URL:

```
http://host:[port]/api/createFolder
```

2. Übergabe Parameter

```
{"active": true, "buzzwords": "Schlagwörter ", "externalKey": "Schlüssel", "foldername": " Ordnername", "mainFolder": true}
```

13 Dokumentenvorschau

```
HttpMethod.GET /thumbnail/{DocID}/page/{pageNr}/height/{height}
```

Liefert das Vorschaubild eines bestimmten Dokuments mit der übergebenen DocID und der angegebenen Seitennummer *'pageNr'*. Dabei ist es immer erforderlich, eine bestimmte Qualität der Vorschau mit Hilfe des Parameters *'height'* zu definieren. Diese Funktion benötigt eine HTTP-Basic-Authentifizierung.

1. Resource-URL:

```
http://host:8080/api/thumbnail/15/page/1/height/1080
```

13.1 Parameter

Es gibt drei erforderliche Parameter für diese Funktion.

13.1.1 docID (required)

DocID steht für "Dokumenten-Identifikationsnummer". Sie wird automatisch und fortlaufend nach der Archivierung für jedes archivierte Dokument vergeben. Die ID-Nummer wird direkt in der URL angegeben. Beispielwert:

```
15
```

13.1.2 pageNr (required)

Die Nummer der gewünschten Seite eines angegebenen Dokuments, zu welcher das Vorschaubild geliefert werden soll. Die Nummer der ersten Seite fängt mit dem Wert *'1'* an. Sollte die angegebene Seite nicht existieren, dann liefert die Funktion "HTTP ERROR 404: Not found" zurück.

13.1.3 height (required)

Die Höhe der Vorschau. Die Angabe der Höhe beeinflusst die Qualität der resultierenden Vorschau. Je größer dieser Wert desto höher ist die Qualität der Vorschau.

13.2 Returns

13.2.1 Datei-Stream

Die Vorschau einer bestimmten Seite vom ausgewählten Dokument.

13.2.2 Vorschau-Größe

Die maximale Ausgabegröße (Höhe / height) für ein Thumbnail beträgt 1080 Pixel.

14 Verlinkungen

Bei der Verlinkung können zu einem Dokument beliebige, archivierte Dateien inklusive deren Klassifizierungen hinzugefügt und somit zu einem Vorgang zusammengeführt werden. Die Verlinkungen werden in der Tabelle des ecoDMS Clients als ausklappbare Untereinträge zum Hauptdokument angezeigt. Der Grundeintrag eines jeden verlinkten Dokuments bleibt dabei selbstverständlich weiterhin bestehen. Diese Funktion gleicht einer virtuellen Dokumentenklammer.

14.1 Verlinken

```
HttpRequest.POST /document/{c1DocId}/linkToDocuments
```

c1DocId Entspricht der Klassifizierungs-Id. Diese ist mit Hilfe der folgenden API-Funktion abrufbar:

```
/documentInfo/{docId}
```

Beispiel URL:

```
http://localhost:8180/api/document/7/linkToDocuments
```

Übergabe der Liste von c1DocIds der zu verknüpfenden Dokumente

Beispiel Parameter für Post:

```
[1, 2, 3, 4, 5, 6]
```

14.2 Anzeigen

```
HttpRequest.GET /document/{c1DocId}/readLinkedDocuments
```

Dieser Befehl liefert eine Liste der verlinkten Dokumente.

Beispiel URL:

```
http://localhost:8180/api/document/7/readLinkedDocuments
```

Beispiel Antwort:

```
[1, 2, 3, 4, 5, 6]
```

14.3 Löschen

```
HttpRequest.POST /document/{c1DocId}/removeDocumentLink
```

Dieser Befehl entfernt die Verlinkungen zwischen Dokumenten.

Beispiel URL:

```
http://localhost:8180/api/document/7/removeDocumentLink
```

Beispiel Daten für Post:

```
[1, 2, 3]
```

Nach den Löschen der Verlinkungen kann mit Hilfe des folgenden Befehls das Ergebnis kontrolliert werden.

```
readLinkedDocuments
```

15 LösCHFunktion

Der API REST Service verfügt über eine eingeschränkte Papierkorbfunktion. Dokumente können in den Papierkorb verschoben werden.

So können nicht mehr benötigte Dokumente / Klassifizierungen aus der Standardansicht (Tabelle im ecoDMS Client) entfernt werden. Dabei werden die Dateien nicht gelöscht, sondern lediglich in einem virtuellen Papierkorb abgelegt.

- Die Such- und Filterfunktionen stehen auch im Papierkorb zur Verfügung.
- Die Klassifizierung ist im Papierkorb nicht möglich.
 - Um die Klassifizierung zu ändern müsste das Dokument zunächst mit der Wiederherstellen-Funktion aus dem Papierkorb genommen werden.

15.1 In Papierkorb verschieben

```
HttpRequest.GET document/{clDocId}/moveToTrash
```

Dieser Befehl verschiebt das gewählte Dokument / die Klassifizierung in den Papierkorb.

Beispiel URL:

```
http://localhost:8180/api/document/7/moveToTrash
```

Beispiel Antwort:

```
true
```

15.2 Wiederherstellen

```
HttpRequest.GET /document/{clDocId}/removeFromTrash
```

Dieser Befehl stellt ein Dokument aus dem Papierkorb wieder her.

Die Suche nach gelöschten Dokumenten ist mit Hilfe der folgenden API-Funktion möglich:

```
searchDocumentsExt
```

Beispiel URL:

```
http://localhost:8180/api/document/7/removeFromTrash
```

Beispiel Antwort:

```
true
```

16 Upload-Größe

Standardmäßig ist die Upload-Größe beim ecoDMS API REST Service auf 10 MB pro Datei begrenzt. Ab ecoDMS Version 18.09 (apu) kann die Upload-Größe am ecoDMS Server bei Bedarf angepasst werden.

16.1 Windows

Um die Upload-Größe bei einer Windows-Installation des ecoDMS Servers zu verändern, gehen Sie wie folgt vor:

1. Öffnen Sie die Dienste-Verwaltung von Windows.
2. Stoppen Sie den Dienst "ecoDMS Server 18.09".
3. Öffnen Sie die Datei "ecodms.properties" in folgendem Verzeichnis:

```
C:\Program Files (x86)\ecoDMS GmbH\ecoDMS Server
```

4. Nun können Sie den Eintrag "rest.api.maxUploadSizePerFile" anpassen. Bitte ergänzen Sie die Datei um die beiden Zeilen:

```
rest.api.maxUploadSize=-1
rest.api.maxUploadSizePerFile=104857600
```

Bitte verändern Sie die anderen Werte in der Datei NICHT!

5. Speichern Sie diese Datei im selben Pfad ab.
6. Starten Sie den Dienst "ecoDMS Server 20" wieder.

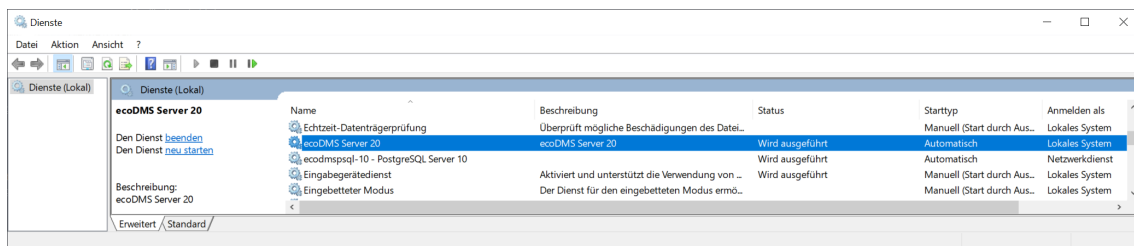


Abb. (ähnlich) 16.1: ecoDMS Server Dienst stoppen unter Windows