



# API Rest Service

ecoDMS Version 14.08 (krusty)

# Impressum

**ecoDMS Version:** 16.09 (eleanor)  
**Handbuch Version:** 11 (1. März 2017)  
**Sprache:** DE

**Ersteller / Urheber:** ecoDMS GmbH  
**©2017** Salierallee 18a  
52066 Aachen  
Deutschland

**Webseite:** [www.ecodms.de](http://www.ecodms.de)  
**E-Mail:** [info@ecodms.de](mailto:info@ecodms.de)  
**Telefon:** 0049 241 47572 01  
**Sitz der Gesellschaft:** Aachen  
**Registergericht:** Amtsgericht Aachen 19201  
**Management:** Dipl.-Betrw. 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, Salierallee 18 a, 52066 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

<b>1</b>	<b>Einleitung</b>	<b>6</b>
1.1	Erklärung Funktionen / Aufruf	6
1.2	API Connects & Lizenzmodell	6
<b>2</b>	<b>API Funktion aktivieren</b>	<b>7</b>
2.1	Dienst starten	7
2.2	Dienst stoppen	7
2.3	Port verwenden	8
2.4	Statistik	8
<b>3</b>	<b>Anmeldung &amp; Autorisierung im ecoDMS API RestService</b>	<b>9</b>
3.1	Verfügbare Funktionen: REST-Commands	9
3.1.1	Verbindung testen	9
3.1.2	Anmeldung und Autorisierung	9
3.2	Parameter	9
3.2.1	archived ID (required)	10
3.2.2	ecoDMSUser (required)	10
3.2.3	ecoDMSPassword (required)	10
3.2.4	Beispiel: Anmeldung und Authentifizierung	10
3.2.5	Verbindung beenden	10
<b>4</b>	<b>Suchfilter</b>	<b>11</b>
4.1	Parameter	11
4.1.1	searchFilterList (required)	11
4.1.2	Beispiel-Werte	12
4.2	Returns	12
4.2.1	docInfoList	12
4.2.2	Beispiel JSON-Liste	12
<b>5</b>	<b>Aufruf der ecoDMS Attribute</b>	<b>13</b>
5.1	Status	13
5.1.1	Returns	13
5.1.1.1	statuslist	13
5.1.1.2	Beispiel JSON-Liste	13
5.2	Rollen / Berechtigungen	13
5.2.1	Returns	13
5.2.1.1	Übersicht der Rollen	13
5.2.1.2	Beispiel Übersicht	14
5.3	Dokumentenarten	14
5.3.1	Returns	14
5.3.1.1	docTypeList	14
5.3.1.2	Beispiel Übersicht	14
5.4	Ordner	14
5.4.1	Returns	15
5.4.1.1	folderList	15
5.4.1.2	Beispiel Übersicht	15
<b>6</b>	<b>Dokumente und Versionen</b>	<b>16</b>
6.1	Parameter	16
6.1.1	docID (required)	16
6.1.2	versionNr (optional)	16

6.1.3	Beispiel REST-Command	17
6.2	Returns	17
6.2.1	documentObject	17
<b>7</b>	<b>Archivierung / Versionierung neuer Dokumente</b>	<b>18</b>
7.1	Datei ins ecoDMS Archiv übertragen	18
7.1.1	Parameter	19
7.1.1.1	file (required)	19
7.1.1.2	versionControlled (required)	19
7.1.2	Returns	19
7.1.2.1	docID	19
7.2	Datei und PDF in ecoDMS Archiv übertragen	19
7.2.1	Parameter	19
7.2.1.1	file (required)	19
7.2.1.2	pdfFile (required)	19
7.2.1.3	versionControlled (required)	20
7.2.2	Returns	20
7.2.2.1	docID	20
7.3	Neue Version hinzufügen	20
7.3.1	Parameter	20
7.3.1.1	file (required)	20
7.3.1.2	docID (required)	20
7.3.1.3	fixed / finalised (required)	20
7.3.2	Returns	20
7.3.2.1	docID	20
7.4	Neue Version mit PDF hinzufügen	21
7.4.1	Parameter	21
7.4.1.1	file (required)	21
7.4.1.2	docID (required)	21
7.4.1.3	fixed (required)	21
7.4.2	Returns	21
7.4.2.1	docID	21
<b>8</b>	<b>Klassifizierung und Dokumenten Informationen</b>	<b>22</b>
8.1	Dokumenten-Informationen ausgeben	22
8.1.1	Parameter	22
8.1.1.1	docID (required)	22
8.1.2	Returns	22
8.1.2.1	docInfoObject	22
8.2	Klassifizierung durchführen	23
8.2.1	Parameter	23
8.2.1.1	docID (required)	23
8.2.1.2	docInfoObject (required)	23
8.2.2	Returns	23
8.2.2.1	docID	23
8.3	Liste definierter Klassifizierungsattribute ausgeben	24
8.3.1	Returns	24
8.3.1.1	classificAttrList	24
8.3.1.2	Beispiel Übersicht	24
8.4	Liste der Standard-Klassifizierungsattribute	24
<b>9</b>	<b>Neuen Ordner erstellen</b>	<b>26</b>
9.1	Parameter	26
9.1.1	dmsOrdnerObject (required)	26
9.1.2	parentoid (optional)	26
9.2	Returns	26
9.2.1	ordnerID	26

<b>10 Vorschau eines Dokuments</b>	<b>27</b>
10.1 Parameter	27
10.1.1 docID (required)	27
10.1.2 pageNr (required)	27
10.1.3 height (required)	27
10.2 Returns	27
10.2.1 Datei-Stream	27
10.2.2 Vorschau-Größe	27

# 1 Einleitung

Speziell für Entwickler gibt es eine Schnittstelle. Der ecoDMS API Rest Service ermöglicht den Anschluss von beliebigen Drittsystemen. So können CRM Systeme, eine Warenwirtschaft oder andere externe Programme via Programmierung direkt an ecoDMS angeschlossen werden. Über die API 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.

## 1.1 Erklärung Funktionen / Aufruf

Über die API 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]
```

## 1.2 API Connects & Lizenzmodell

Um die ecoDMS API nutzen zu können, benötigen Sie API Connects für den Up- und Download.

- 1 API Connect = 1 Upload oder Download über die ecoDMS API pro Monat
- API Connects sind zeitlich unbegrenzt für die dazugehörige ecoDMS Version nutzbar und können frei nach Kundenwunsch erworben werden.
- Damit Sie die API testen können, erwerben Sie automatisch mit jeder ecoDMS Lizenz monatlich 10 API Connects.
  - Diese bleiben auch beim Kauf weiterer API Connects bestehen.
  - Während der 30-tägigen Demophase sind in ecoDMS ebenfalls 30 API Connects für Sie freigeschaltet.
- Monatlich werden die API Connects automatisch zurückgesetzt, so dass Ihnen jeden Monat die volle Anzahl erworbener API Connects zur Verfügung steht.
- Zur Verwendung der API wird eine freie, gleichzeitige Verbindung auf das ecoDMS Archiv benötigt.
- Gezählt werden alle Up- und Downloads.
  - Der Abruf von beispielsweise Ordnerlisten wird selbstverständlich nicht als Connect abgezogen.
- Der aktuelle Stand der monatlich zur Verfügung stehenden API Connects kann über die API selbst und über den ecoDMS Client (Einstellungsdiallog) abgefragt werden.

## 2 API Funktion aktivieren

Um die ecoDMS API einsetzen zu können, muss der Service zunächst aktiviert werden.

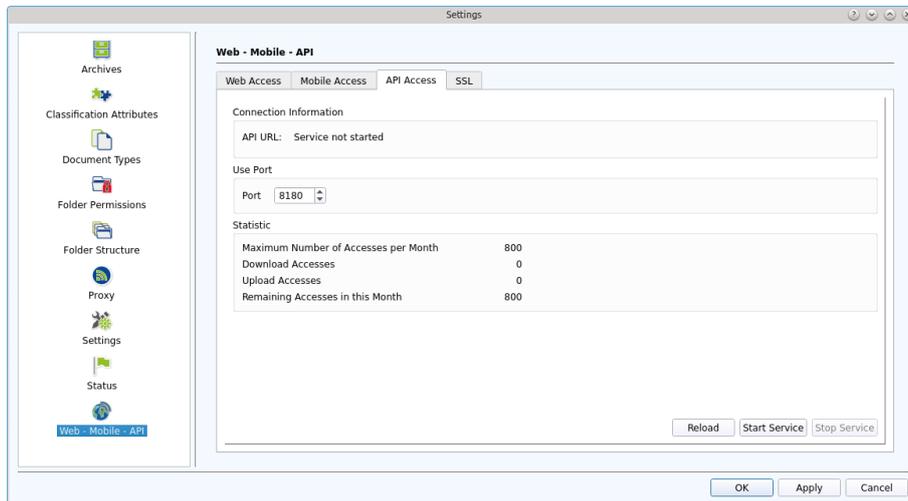


Abb. (ähnlich) 2.1: Einstellungen - Web - Mobil - API - API Zugriff

### 2.1 Dienst starten

Um die Schnittstelle (API) nutzen zu können, müssen Sie zunächst den Dienst aktivieren. Gehen Sie wie folgt vor:

1. Melden Sie sich als ecoDMS Administrator an und starten Sie den Einstellungsdialog im ecoDMS Client.
2. Wählen Sie im Einstellungsdialog den Reiter "Web - Mobil - API -> API Zugriff" aus.
3. Klicken Sie zum Starten des Dienstes auf den Button "Dienst starten".
  - a) Der Dienst wird gestartet. Dieser Prozess kann etwas dauern.
4. Die notwendige "API URL" wird im Bereich "Verbindungsinformationen" angezeigt.
5. Speichern Sie den Vorgang mit einem Klick auf "Anwenden" im Einstellungsdialog.

### 2.2 Dienst stoppen

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

1. Melden Sie sich als ecoDMS Administrator an und starten Sie den Einstellungsdialog im ecoDMS Client.
2. Wählen Sie im Einstellungsdialog den Reiter "Web - Mobil - API -> API Zugriff" aus.
3. Klicken Sie zum Stoppen des Dienstes auf den Button "Dienst stoppen".
  - a) Der Dienst wird gestoppt. Dieser Prozess kann etwas dauern.
4. Speichern Sie den Vorgang mit einem Klick auf "Anwenden" im Einstellungsdialog.

## 2.3 Port verwenden

Standardmäßig verwendet die API Port "8180". Den Port können Sie wie folgt ändern:

1. Melden Sie sich als ecoDMS Administrator an und starten Sie den Einstellungsdialog im ecoDMS Client.
2. Wählen Sie im Einstellungsdialog den Reiter "Web - Mobil - API -> API Zugriff" aus.
3. Geben Sie den Port in das Feld "Port" ein.
4. Speichern Sie den Vorgang mit einem Klick auf "Anwenden" im Einstellungsdialog.

## 2.4 Statistik

Um die API nutzen können, werden monatliche API Connects (Lizenzen) benötigt. Der aktuelle Stand der zur Verfügung stehenden Einheiten kann abgerufen werden.

1. Melden Sie sich als ecoDMS Administrator an und starten Sie den Einstellungsdialog im ecoDMS Client.
2. Wählen Sie im Einstellungsdialog den Reiter "Web - Mobil - API -> API Zugriff" aus.

Folgende Informationen stehen im Bereich "Statistik" zur Verfügung:

1. **Maximale Zugriffe pro Monat:** Erworbene / Verfügbare Anzahl an monatlichen API Connects (Lizenzen).
2. **Download Zugriffe:** Anzahl der bereits abgerufenen bzw. heruntergeladenen Dokumente über die API im aktuellen Monat.
3. **Upload Zugriffe:** Anzahl der bereits archivierten bzw. hochgeladenen Dokumente über die API im aktuellen Monat.
4. **Verbleibende Zugriffe in diesem Monat:** Anzahl der verbleibenden API Connects (Lizenzen) im aktuellen Monat.

## 3 Anmeldung & Autorisierung im ecoDMS API RestService

API RestService 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.

### 3.1 Verfügbare Funktionen: REST-Commands

#### 3.1.1 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 "applord Test: OK".

#### 3.1.2 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 API Rest Service Benutzer nicht mit den gültigen ecoDMS-Benutzerdaten autorisiert hat (s. Bsp. Anmeldung und Autorisierung).

1. Resource URL:

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

### 3.2 Parameter

Es gibt verschiedene Parameter, die verwendet werden können.

### 3.2.1 archived ID (required)

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

1. Beispiel-Wert:

```
1
```

### 3.2.2 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
```

### 3.2.3 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
```

### 3.2.4 Beispiel: Anmeldung und Authentifizierung

```
musteruser:mypassword
```

Ausgabe Base64 codiert (HTTP Basic Authentifizierung)

```
Basic bXVzdGVydXNlcjpteXBhc3N3b3Jk
```

### 3.2.5 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 successfull
```

## 4 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
```

### 4.1 Parameter

Es gibt verschiedene Parameter, die für die Suche verwendet werden können.

#### 4.1.1 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.

## 4.1.2 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"}]
```

## 4.2 Returns

### 4.2.1 docInfoList

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

### 4.2.2 Beispiel JSON-Liste

```
[  
{  
  "DocID":15, "clDocId":15, "archiveName": "1", "classifyAttributes":  
  {"changeid": "ecoDMS", "status": "3", "mainfolder": "1", "ctimestamp": "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", "ctimestamp": "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}  
}]
```

## 5 Aufruf der ecoDMS Attribute

Jedes einzelne ecoDMS Klassifizierungsattribut kann aufgerufen werden.

### 5.1 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
```

#### 5.1.1 Returns

##### 5.1.1.1 statuslist

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

##### 5.1.1.2 Beispiel JSON-Liste

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

### 5.2 Rollen / Berechtigungen

```
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
```

#### 5.2.1 Returns

##### 5.2.1.1 Übersicht der Rollen

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

### 5.2.1.2 Beispiel Übersicht

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

## 5.3 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
```

### 5.3.1 Returns

#### 5.3.1.1 docTypeList

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

#### 5.3.1.2 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}}
]
```

## 5.4 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
```

## 5.4.1 Returns

### 5.4.1.1 folderList

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

### 5.4.1.2 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":""}]
```

## 6 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 im ecoDMS archivierte Dokument repräsentiert. Jedes ecoDMS File-Objekt besitzt die folgende Struktur:

<b>docid</b>	38
Dateigröße	98890
Dokumentenname	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
```

### 6.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

#### 6.1.1 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
```

#### 6.1.2 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
```

### 6.1.3 Beispiel REST-Command

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

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

## 6.2 Returns

### 6.2.1 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.

# 7 Archivierung / Versionierung neuer Dokumente

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

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

1. `/uploadFile`

Als Erstes 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 Dokumente erfolgreich auf dem ecoDMS Server hochgeladen wurde und eine entsprechende DocID vorliegt, kann die Datei optional zunächst klassifiziert werden. Das Setzen der Werte ist aber nicht unbedingt erforderlich. 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 deren 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.

## 7.1 Datei ins ecoDMS Archiv übertragen

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

Diese Funktion überträgt eine neue Datei ins ecoDMS Archiv 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
```

### 7.1.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

#### 7.1.1.1 file (required)

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

#### 7.1.1.2 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

### 7.1.2 Returns

#### 7.1.2.1 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.

## 7.2 Datei und PDF in ecoDMS Archiv übertragen

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

Diese Funktion überträgt eine neue Datei und das dazugehörige PDF ins ecoDMS Archiv. 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
```

### 7.2.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

#### 7.2.1.1 file (required)

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

#### 7.2.1.2 pdfFile (required)

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

### 7.2.1.3 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

## 7.2.2 Returns

### 7.2.2.1 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.

## 7.3 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
```

### 7.3.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

#### 7.3.1.1 file (required)

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

#### 7.3.1.2 docID (required)

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

```
15
```

#### 7.3.1.3 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

### 7.3.2 Returns

#### 7.3.2.1 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.

## 7.4 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
```

### 7.4.1 Parameter

Es gibt verschiedene Parameter für diese Funktion.

#### 7.4.1.1 file (required)

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

#### 7.4.1.2 docID (required)

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

- Zum Beispiel : PDF

#### 7.4.1.3 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

### 7.4.2 Returns

#### 7.4.2.1 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.

## 8 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.

### 8.1 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
```

#### 8.1.1 Parameter

Es gibt folgende Parameter für diese Funktion.

##### 8.1.1.1 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
```

#### 8.1.2 Returns

##### 8.1.2.1 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}
```

## 8.2 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
```

### 8.2.1 Parameter

Es gibt folgende Parameter für diese Funktion.

#### 8.2.1.1 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
```

#### 8.2.1.2 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}
```

### 8.2.2 Returns

#### 8.2.2.1 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.

## 8.3 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
```

### 8.3.1 Returns

#### 8.3.1.1 classificAttrList

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

#### 8.3.1.2 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"
}
```

## 8.4 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";
```

## 9 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
```

### 9.1 Parameter

Es gibt hierbei folgende Parameter:

#### 9.1.1 dmsOrdnerObject (required)

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

#### 9.1.2 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
```

### 9.2 Returns

#### 9.2.1 ordnerID

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

# 10 Vorschau eines Dokuments

```
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/500
```

## 10.1 Parameter

Es gibt drei erforderliche Parameter für diese Funktion.

### 10.1.1 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
```

### 10.1.2 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.

### 10.1.3 height (required)

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

## 10.2 Returns

### 10.2.1 Datei-Stream

Das Vorschaubild einer bestimmten Seite vom ausgewählten Dokument.

### 10.2.2 Vorschau-Größe

Die maximale Ausgabegröße (Höhe / height) für ein Thumbnail beträgt 500 Pixel.