



USM 2.0

JSON-API Spezifikation
Version 2.1.1

Einleitung	7
Grundlagen	8
Response Status Codes	9
USM-JSON-API (Version 1.6.x)	11
Login	11
Logout	13
Get Configuration	14
Set Configuration	16
Synchronisations-System allgemein	17
UUIDs	18
ConflictResolution	18
Attachments zu PIM-Objekten	19
Sync-Fehler-Rückgabe-Daten	19
Fehlermeldungen für einzelne Elemente	20
Sync Init	22
Sync Update	24
Sync Estimate	26
Set Push Notification	27
Get Cached Data	28
Read Data	29
Send Mail	30
Resolve Recipients	31
Move Items	32
Ping	34
Resolve OX IDs	35
Invalidate Objects	36
Locate Objects	37
Set Confirmation Status	38
Erweiterungen USM-JSON-API (Version 2.0.0)	39
Anwendungsmöglichkeiten der Ersetzung mittels „tempid“	40
Einschränkungen bei der Nutzung von „tempid“-Daten	40
Download Data	41
Upload Data	42
Clear Data	44
Erweiterungen USM-JSON-API (Version 2.1.0)	45
OX-JSON-CALLS	45
Modul: folders	46
Get root folders	46
Get subfolders	46
Get path	47

Get updated folders	47
Get a folder	48
Create a folder	48
Update a folder	49
Delete folders	49
Modul: tasks	50
Get all tasks	50
Get a list of tasks	51
Get updated tasks	51
Get a task	52
Create a task	52
Update a task	53
Delete tasks	53
Confirm a task	54
Search for tasks	54
Search Tasks Object Fields	55
Modul: contacts	55
Get all contacts	55
Get a list of contacts	56
Get a list of users	57
Get updated contacts	57
Get a contact	58
Get a contact by user ID	58
Create a contact	59
Update a contact	59
Delete contacts	60
Search contacts	60
Modul: calendar	63
Get all appointments	63
Get a list of appointments	63
Get updated appointments	64
Get appointment information	65
Get an appointment	65
Create an appointment	66
Update an appointment	66
Delete appointments	67
Confirm appointment	68
Search appointments	68
Get new appointments	69
Free & Busy	70
Modul: mail	71
Get mail count	71
Get all mails	71
Search mails	72
Get a list of mails	73
Copy mails	74
Move mails	74
Update mails	75
Get a mail	75
Get a mail attachment	76
Send / Save mail	77
Reply / Forward a mail	78
Delete mails	79
Clear mail folders	79
Module: groups	80
Get a group	80
List groups	80

Search for groups.....	81
Create a group.....	81
Delete a group.....	82
Change a group.....	82
Modul: resource	83
List resources	83
Get a resource	84
Search for resources	84
Module: infostore.....	85
Get all infoitems	85
Get a list of infoitems.....	86
Get updated infoitems	86
Get an infoitem	87
Search infoitems	88
Get an infoitem document	89
Get all versions	89
Update an infoitem.....	90
Create an infoitem.....	91
Save an attachment in the infostore	91
Delete infoitems.....	92
Delete versions of infostore documents	92
Delete all versions of infostore documents	93
Copy an infostore document.....	94
Lock an infoitem.....	94
Unlock an infoitem	95
Modul: Attachments	95
Get a list of attachments	96
Create an Attachment.....	97
Delete Attachment	98
Get updated attachments	98
Get an Attachment	99
Get an attachments filedata	100
Modul: reminder.....	102
Get a reminder range.....	102
Delete Reminder	103
Modul: multiple.....	104
Multiple requests	104
Modul: quota.....	106
Get the filestore usage data	106
Get the mail usage data	106
Modul: import.....	107
Import CSV	107
Import Outlook CSV.....	107
Import iCAL	108
Import vCard	109
Modul: export.....	110
Exporting CSV	110
Exporting iCAL	110
Exporting vCAD	111
Modul: sync	112
Clearing a folder's content	112
Modul: mailfilter	112
Modul: ajax file upload	112

Uploading a file	112
Uploading a file's last access timestamp (keep alive)	113
Requesting a formerly uploaded a file	113
Modul: image	114
Requesting an image	114
Modul: coversion (preliminary)	114
Converting data.....	114
Saving a vCard email attachment	116
Modul: mail account (preliminary)	117
Get a mail account	117
Create a new mail account.....	119
Upadate a mail account	119
Delete a mail account.....	120
Validate a mail account (which shall be created)	120
Anhang A: Liste der USM Error Codes	121
Configuration Error Codes:.....	121
Content Type Error Codes:.....	121
Database:	123
Mapping:	124
USM Session Management:.....	124
JSON Error Codes:.....	125
Toolkit (Conversion errors):	126
Anhang B USM ContentTypes	126
FolderContentType.....	126
Common folder data	126
Detailed folder data.....	127
Permission flags.....	128
Permission object.....	128
Capabilities	129
Common object data.....	129
TaskContentType.....	130
Identifikator	130
ContentType Daten	130
Detailed task and appointment data	130
Participant identifier.....	132
User participant object	132
Detailed task data.....	133
AppointmentContentType.....	134
Identifikator	134
ContentType Daten	134
Detailed appointment data.....	134
ContactContentType.....	136
Identifikator	136
ContentType Daten	136
Detailed contact data.....	136
Distribution list member	140
MailContentType	140
Identifikator	140
ContentType Daten	140

Detailed mail data	141
Simple Mail Body	142
Multipart Mail Body	142
GroupContentType	142
Identifikator	142
ContentType Daten	142
Group data	143
ResourcesContentType	143
Identifikator	143
ContentType Daten	143
Anhang C: OX Content Types	143
InfoitemContentType	143
Identifikator	143
ContentType Daten	143
Detailed infoitem data	143

Einleitung

Gegenstand dieses Dokuments ist die Definition der USM-JSON Schnittstelle, die unter anderem für die Anbindung von Outlook Clients an einen Open-Xchange Server benutzt werden kann.

Das erste Kapitel beschreibt Grundsätze und Definitionen, die für die gesamte Schnittstelle gelten.

Im zweiten Kapitel werden die einzelnen JSON-Befehle beschrieben, die von USM zur Anbindung von Clients und zur Synchronisation von Daten zur Verfügung gestellt werden.

Das dritte Kapitel enthält eine Auflistung weiterer Befehle, die in zukünftigen Versionen möglicherweise vom USM als Proxy an den OX-Server weitergeleitet werden.

Anhang A enthält eine Liste allen von USM gelieferten Standard-Fehler Codes. Diese Liste wird ggfs. Im Laufe des Projekts um weitere spezifische Codes erweitert.

Anhang B enthält die Beschreibung aller aktuell im USM definierten Standard-ContentTypes.

Anhang C enthält die OX ContentTypes, die im USM nicht abgebildet sind, aber für bestimmte mögliche zukünftige OX-Proxy-Calls relevant sind.

Grundlagen

Die USM-JSON-API arbeitet über das HTTP(S)-Protokoll. Die Befehle werden im Normalfall als HTTP PUT Requests gesendet (Ausnahmen werden bei dem entsprechenden Befehl beschrieben). Als Text-Encoding wird sowohl bei den Request-Parametern als auch in der Response das UTF-8-Encoding erwartet bzw. benutzt.

Die Request-URL jedes Befehls besteht aus Basis-Pfad für die USM-JSON-Schnittstelle, gefolgt vom Befehls-Namen als Sub-Pfad. Die Parameter des Befehls werden immer im Request-Body in einem JSONObject erwartet.

Um effizientes Caching und Load-Balancing zu ermöglichen, kann der USM-Server Cookies in der Antwort übergeben. Soweit irgendwie möglich, sollte ein Client bei zukünftigen Befehlen für die gleiche Session dieses Cookie mitsenden, um eine effiziente Ausführung zu gewährleisten.

Die Response zu einem USM-JSON-Befehl ist immer im Content-Type „text/javascript“ gehalten. Sie enthält immer ein JSONObject mit einem fest vorgegebenem Feld namens „status“, das einen Integer-Wert enthält. In diesem Feld wird der allgemeine Return-Code des Befehls übermittelt. Abhängig vom Inhalt des Status-Felds sind weitere Felder im Ergebnis-Objekt gesetzt:

- 001 (OK):
 - „data“: enthält das spezifische Ergebnis des Befehls. Dies kann ein JSONObject, ein JSONArray als auch jeder andere in JSON gültige primitive Datentyp sein. Bei den einzelnen Befehlen ist jeweils aufgelistet, welche Daten zurückgeliefert werden.
- Sonstige:
 - „errorCode“ : (Integer) USM Fehler Code. Alle Codes sind im Anhang A aufgelistet.
 - „errorMessage“: (String) Beschreibung des Fehlers in englischem Klartext
 - „errorDetails“: (optional, JSONObject) Enthält fehlerspezifische weiterführende Informationen.
 - „oxError“: (optional, JSONObject) Enthält vom OX-Server gelieferte Fehlerdetails, falls ein unerwarteter OX-Fehler während der Abarbeitung des Befehls aufgetreten ist.

Der Status-Code ist unabhängig vom Befehl und beschreibt im Falle eines Fehlers das allgemeine Problem, das diesen Fehler verursacht hat. Alle Status Codes sind im Folgenden aufgelistet:

Response Status Codes

Im folgenden findet sich eine Liste aller Status Codes, die in dem Response zurückgeliefert werden. Zu jedem Status-Code ist ein Klartextbezeichner und eine kurze Beschreibung angegeben:

Status-Code	Bezeichnung	Beschreibung
1	Success	Der Befehl wurde erfolgreich ausgeführt
2	OX-Server Error	Es trat ein unerwarteter Fehler beim Zugriff auf OX auf
3	Database Error	Es trat ein Fehler beim Zugriff auf die Datenbank auf
4	Internal Error	Es trat ein unerwarteter interner Fehler im USM auf
5	Access Denied	Eine Anmeldung an USM/OX war nicht erfolgreich
6	Unknown Session	Die übergebene Session-ID ist unbekannt
7	Unknown UUID	Eine übergebene UUID ist USM nicht bekannt
8	Wrong/Missing Parameters	Es fehlen notwendige Parameter zur Ausführung des Befehls oder es wurden ungültige Werte übergeben
9	Sync Failed	Eine Synchronisation ist aufgrund von Konflikten fehlgeschlagen. Im Feld „errorDetails“ sind Informationen über die Konflikte gespeichert
10	Bad Request	Der angeforderte Befehl ist dem USM-Server nicht bekannt
11	Unknown Syncid	Vom Client wurde eine Sync-ID übergeben, die dem Server nicht bekannt ist. Der Client muss den Vorgang mit einer gültigen Sync-ID wiederholen oder einen Syncinit für den Folder oder die Folder-Hierarchie durchführen
12	Unsupported Version	Die vom Client beim Login angefragte Version wird vom USM-Server nicht unterstützt. In den Details sind die angefragte („requested“) und verfügbare („available“) Version enthalten. Der Client muss eine passende Version verwenden (was entweder über die Benutzung einer passenden Version oder über einen Upgrade des Clients oder Servers zu erreichen ist)
13	Unknown Folder	Der zu synchronisierende Folder ist auf dem Server nicht mehr vorhanden. Der Client muss eine Synchronisierung der Folder-Hierarchie durchführen.
14	Unknown UID	Eine übergebene UID konnte vom USM auf dem Server nicht gefunden werden.
15	Unallowed local time	Die Client-Zeit weicht mindestens eine Stunde von der Server-Zeit ab.
16	OX-Server/Resource temporary not available	Ein Resource/Subsystem ist verübergehend nicht erreichbar. Entspricht Fehler Kategorie: 4,5,6 und 11 vom OX Server.
17	Success (Filtered out large objects)	Die Synchronisation war erfolgreich, im Folder auf dem Server befinden sich allerdings Elemente, welche die konfigurierte maximale Größe überschreiten und deshalb ausgefiltert werden (d.h. für den Client nicht sichtbar sind)

Diese Liste kann noch um weitere Status-Codes erweitert werden, ein Client sollte ihm unbekannte Status-Codes analog wie allgemeine Fehler (Codes 2-4) behandeln.

Jeder Befehl (außer Login) kann als Status-Code 6 („Unknown Session“) zurückliefern, in diesem Fall muss sich ein Client neu anmelden und den Befehl dann wiederholen.

USM-JSON-API (Version 1.6.x)

Login

Pfad:

<Basis-URL>/login

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
user	String	Nein		
device	String	Nein		
password	String	Nein		
configuration	Boolean	Ja	false	Wenn true, werden Standard-Konfigurations-Felder wie in GetConfiguration beschrieben geliefert
availableContentTypes	Boolean	Ja	false	Siehe GetConfiguration
syncContentTypes	Boolean	Ja	false	Siehe GetConfiguration
version	String	Ja	“	Vom Client angeforderte Version der JSON-API
local_time	String	Ja		Die lokale Client-Zeit im format "yyyy-MM-dd HH:mm:ss.SSS"

Response:

JSONObject:

Feld	Datentyp	Optional	Beschreibung
sessionid	String	Nein	Identifikator für die Session für andere Befehle
configuration	JSONObject	Ja	Teile der Konfiguration wie über Parameter bestimmt
version	String	Nein	Vom USM-Server zur Verfügung gestellte Version der JSON-API
userid	Integer	Nein	OX-User-ID des angemeldeten Users, wird für einige Felder (z.B. Eigentümer von Objekten) verwendet
usmVersion	String	Nein	Versionsnummer USM, z.B. 6.18.0
usmBuild	String	Nein	SVN-Build-Number von USM
usmJSONVersion	String	Nein	Versionsnummer des USM-JSON-Bundles

Beschreibung:

Dieser Befehl wird zur Authentifizierung eines Clients genutzt. Als Parameter werden der Identifikator des Benutzers („user“), sein Passwort („password“) und ein eindeutiger Identifikator für das Client-Device („device“) erwartet. Über den Benutzer und das Device wird vom USM eine Session eindeutig bestimmt.

Über die optionalen Parameter „configuration“, „availableContentTypes“ und „syncContentTypes“ kann vom Client spezifiziert werden, ob bei einem erfolgreichen Login Konfigurationseinstellungen vom USM mitgeliefert werden sollen.

Für den Fall einer erfolgreichen Anmeldung liefert das USM als Ergebnis einen Identifikator („sessionid“) für die Session zurück, welcher bei allen weiteren JSON-Befehlen übergeben werden muss. Ist einer der optionalen Parameter auf „true“ gesetzt, wird im Ergebnisfeld „configuration“ zusätzlich die aktuelle Konfiguration der Session zurückgeliefert. Die Struktur entspricht dabei der Antwort des GetConfiguration-Befehls, der Parameter „configuration“ legt fest, ob die Standard-Felder geliefert werden sollen.

Beim Login kann der Client eine gewünschte Version der USM-JSON-API übergeben. Eine Version muss dabei folgendem Format entsprechen:

[0-9]+(\.[0-9]+)([]+.)?

Dies bedeutet, dass eine Version mit einer positiven Integer beginnen muss, gefolgt von einem oder mehr Sub-Versions-Nummern, die ebenfalls positive Integer sein müssen und durch einen Punkt von höherwertigen Anteilen der Version getrennt sind. Nach der Versionsnummer darf getrennt durch mind. 1 Leerzeichen ein beliebiger Text folgen, der als zusätzliche Beschreibung angegeben werden kann. Beispiele:

```
1.0.0 Initial Release
1.0.1
1.1.0 With additional configuration information
2.0.0 New API Structure
2.0.1 Bugfix #1555
```

Vom USM werden nur die ersten 3 Versions-Teile beachtet, alle weiteren werden zur Bestimmung einer kompatiblen Version ignoriert, ebenso die Beschreibung der Version. Als Konvention werden folgende Annahmen getroffen:

1. Änderungen im ersten Versions-Teil stellen inkompatible Änderungen dar (z.B. neue Struktur, andere Befehle).
2. Änderungen im zweiten Versions-Teil stellen Erweiterungen dar, die mind. Auf Serverseite so implementiert wurden, dass sie bei Nichtangabe kompatibel zu Vorgängerversionen sind (z.B. durch Verwendung passender Default-Werte).
3. Änderungen im dritten Versions-Teil stellen kleinere Anpassungen/Fixes dar, die im Allgemeinen nur interner Natur sind und keine Änderung in der API zur Folge haben.

Zur Ermittlung einer kompatiblen Version werden folgende Überprüfungen durchgeführt:

1. Der USM-Server muss eine Version zur Verfügung stellen, die im ersten Teil genau mit der vom Client angeforderten Version übereinstimmt.
2. Der USM-Server muss eine Version zur Verfügung stellen, die im zweiten Teil mindestens die entsprechende Nummer der vom Client geforderten Version enthält, es darf auch eine größere Nummer genutzt werden

Wenn beide Bedingungen erfüllt sind, wird bei der Antwort des Login-Befehls die vom USM-Server verwendete Version zurückgemeldet. Wenn der USM-Server keine passende USM-Version ermitteln kann, wird als Status-Code 12 (Unsupported Version) zurückgeliefert. Sollte der Client keine gültige Version (d.h. nicht dem obigen Muster entsprechend) übergeben, wird als Status-Code 8 (Wrong/Missing Parameters) zurückgeliefert. Wenn der Client keine

Versionsnummer als Parameter übergibt, liefert der USM-Server die aktuellste Version zurück.

Beispiel:

Request:

```
PUT http://localhost/usm/login
Content-Type: text/javascript

{"user":"user2", "device":"device1", "password":"pass" }
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1, "data": {"sessionid": "USM-1","userid":
12345,"version": "1.0.0" } }
```

Logout

Pfad:

<Basis-URL>/logout

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
endSynchronization	Boolean	Ja	false	

Response:

Leer

Beschreibung:

Mit diesem Aufruf meldet sich ein Client beim USM-System ab. Als Parameter muss die zuvor über den Login-Befehl erhaltene Session-ID übergeben werden. Im Normalfall meldet sich ein Client damit nur temporär vom USM-System ab (so dass auf dem Server temporäre Ressourcen freigegeben werden können), alle Synchronisationsinformationen sowie die Konfigurationseinstellungen sind persistent gespeichert. Wird der optionale Parameter „endSynchronization“ mit dem Wert „true“ mit übergeben, werden vom USM allerdings zusätzlich alle persistenten Daten gelöscht, dieser Parameter darf also nur gesetzt werden, wenn die Synchronisation zwischen Client und Server endgültig beendet werden soll.

Beispiel:

Request:

```
PUT http://localhost/usm/logout
```

```
Content-Type: text/javascript

{"sessionId":"USM-1"}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1}
```

Get Configuration

Pfad:

<Basis-URL>/getConfiguration

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionId	String	Nein		
availableContentTypes	Boolean	Ja	false	Wenn auf true gesetzt wird in der Response eine Liste aller verfügbarer ContentTypes zurückgeliefert
syncContentTypes	Boolean	Ja	false	Wenn auf true gesetzt wird in der Response eine Liste aller synchronisierten ContentTypes mit ihren jeweils synchronisierten Feldern zurückgeliefert

Response:

JSONObject:

Feld	Datentyp	Optional	Beschreibung
context_uuid	String	Nein	UUID des Contexts des OX-Servers, in dem der Benutzer registriert ist
context_id	Integer	Nein	OX-ID des Contexts des OX-Servers, in dem der Benutzer registriert ist
defaultAddress	String	Nein	Default Email Adresse des Benutzers für diese Session
user	String	Nein	Identifikator des Benutzers für diese Session
device	String	Nein	Client-Device-Identifikator für diese Session
timezone	String	Nein	In OX eingestellte Zeitzone des Users
tree	Integer	Nein	Identifikator des Folder-Trees des Users
startDate	Long	Nein	Beginn des Synchronisationszeitraums (Millisekunden seit dem 1.1.1970 0:00 GMT)
endDate	Long	Nein	Ende des Synchronisationszeitraums

			(Millisekunden seit dem 1.1.1970 0:00 GMT)
conflictResolution	String	Nein	Default-Konflikt-Lösungsstrategie
customProperties	JSONObject	Nein	Frei setzbare persistente Felder des Clients
availableContentTypes	JSONObject	Ja	Liste aller für die Synchronisation verfügbaren ContentTypes und Ihrer Felder
syncContentTypes	JSONObject	Ja	Liste aller für die Synchronisation verwendeten ContentTypes und Ihre synchronisierten Felder
userid	Integer	Nein	OX-User-ID des angemeldeten Users, wird für einige Felder (z.B. Eigentümer von Objekten) verwendet

Beschreibung:

Dieser Befehl liefert die aktuell auf dem USM eingestellte Konfiguration für diese Session zurück. Als Parameter muss die zuvor über den Login-Befehl erhaltene Session-ID übergeben werden.

Zusätzlich zu den Standard-Feldern werden Custom-Properties geliefert, dies sind persistente Key/Value-Kombinationen, die vom Client in früheren SetConfiguration-Befehlen gespeichert wurden. Für die persistenten Custom-Properties gilt, dass sowohl Key als auch Value jeweils ein String sein muss, der Key darf eine Länge von 13 Zeichen nicht überschreiten, die Value kann maximal 32 Zeichen enthalten.

Ist der optionale Parameter „availableContentTypes“ auf „true“ gesetzt, wird im gleichnamigen Feld ein JSONObject geliefert, welches als Key-Value-Kombinationen jeweils den Bezeichner eines ContentTypes und ein JSONArray mit allen von ihm zur Verfügung gestellten Feldern enthält.

Im Feld „syncContentTypes“ wird ebenfalls ein gleichartig aufgebautes JSONObject geliefert, falls der gleichnamige Parameter auf „true“ gesetzt war. Hier werden jedoch nicht alle zur Verfügung stehenden ContentTypes gelistet, sondern diejenigen, für die eine Synchronisation aktiviert ist. Die im jeweils zugehörigen JSONArray gelieferten Feld-Namen sind die Felder des ContentTypes, die bei der Synchronisation berücksichtigt werden sollen.

Für eine neu initialisierte Session sind per Default alle ContentTypes mit allen von Ihnen zur Verfügung gestellten Feldern für die Synchronisation eingetragen.

Beispiel:

Request:

```
PUT http://localhost/usm/getConfiguration
Content-Type: text/javascript

{"sessionid":"USM-1"}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1,"data":{"context_uuid": "c16ab579-0327-4de8-bb9f-dabb432bcad7", "user":"user1", "userid":12345, "device":"device1", "timezone":"Europe/Berlin", "tree":
"1", "startDate":1252707997780, "endDate":1253022148180,
"conflictResolution":"USE_SERVER", "customProperties":{}}}
```

Set Configuration

Pfad:

<Basis-URL>/setConfiguration

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionId	String	Nein		
syncContentTypes	JSONObject	Ja		Wenn gesetzt werden die übermittelten Felder der ContentTypes für zukünftige Synchronisationen als relevant markiert
startDate	Long	Ja		Siehe GetConfiguration
endDate	Long	Ja		Siehe GetConfiguration
tree	Integer	Ja		Siehe GetConfiguration
conflictResolution	String	Ja		Siehe GetConfiguration
customProperties	JSONObject	Ja		Gesetzte Custom-Properties werden persistent gespeichert

Response:

Leer

Beschreibung:

Dieser Befehl ändert die aktuell auf dem USM eingestellte Konfiguration für diese Session. Als Parameter muss die zuvor über den Login-Befehl erhaltene Session-ID übergeben werden.

Ist ein beliebiger der optionalen Parameter nicht gesetzt, werden die Werte in der Konfiguration für dieses Feld nicht geändert.

Wenn der optionale Parameter „syncContentTypes“ gesetzt ist, werden für alle in ihm spezifizierten ContentTypes die Felder, die synchronisiert werden, auf die als JSONArray übergebene Liste beschränkt. ContentTypes, die in diesem JSONObject nicht übergeben wurden, werden nicht verändert. Dies bedeutet insbesondere, dass man eine leere Feldliste für einen ContentType übergeben muss, um diesen komplett von der Synchronisation auszuschliessen. Für den Parameter „syncContentTypes“ können auch die Folgende Spezialfälle übergeben werden:

1. "all" (String):

Alle bereitgestellten Content-Types werden mit allen Feldern synchronisiert

2. JSONObject mit Content-Types als Keys und Value "all" (String):

Für den angegebenen Content-Type werden alle Felder synchronisiert

3. JSONObject mit Content-Types als Keys und Value JSONArray mit einer Liste deren erstem Element "all except" ist (String):

Für den angegebenen Content-Type werden alle Felder, außer die aufgelisteten, synchronisiert.

Dabei werden vom USM immer die aktuell registrierten Content-Types bzw. ihre Felder zur Synchronisierung eingetragen. Werden später Content-Types oder Felder eines Content-Types hinzugefügt, werden diese nicht automatisch nachträglich zur Synchronisation hinzugefügt (gilt für beide Varianten).

Gleiches (außer die Spezialfälle mit „all“) gilt für den optionalen Parameter „customProperties“: Es werden nur die Properties verändert, die explizit in dem zugehörigen JSONObject übergeben wurden. Um eine Custom-Property zu löschen, muss diese explizit auf einen leeren String gesetzt werden.

Beispiel:

Request:

```
PUT http://localhost/usm/setConfiguration
Content-Type: text/javascript

{"sessionId":"USM-1","conflictResolution":"USE_CLIENT",
"customProperties":{"propertySet":"set","propertyClear":""}}

PUT http://localhost/usm/setConfiguration
Content-Type: text/javascript

{"sessionId":"id","syncContentTypes":"all"}
-> Aktiviert alle Felder aller aktuell registrierten Content-
Types für die Synchronisation

PUT http://localhost/usm/setConfiguration
Content-Type: text/javascript

{"sessionId":"id","syncContentTypes":{"mail":"all",
"contacts":[],"calendar":["id","folder_id","title"]}}
-> Aktiviert all Mail-Felder, deaktiviert Kontakte, und
beschränkt Termine auf 3 Felder
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1}
```

Synchronisations-System allgemein

Das Sync-System von USM baut auf zu synchronisierenden Objekten (im folgenden „DataObject“ genannt) und in ihnen enthaltenen Feldern (im folgenden „Properties“ genannt) auf. Jedes DataObject ist fest einem ContentType zugeordnet und kann genau die Properties

enthalten, die von diesem ContentType zur Verfügung gestellt werden. Jede Property hat intern einen Datentyp, dies kann ein primitiver Typ („String“, „Integer“,...) oder auch ein komplexer Datentyp (z.B. „Array von Strings“) sein.

In der JSON-API von USM wird ein DataObject durch ein JSONObject repräsentiert. Jede Property ist in diesem JSONObject mit ihrem Namen als Key hinterlegt. Der Wert der Property ist als Value zu diesem Key im JSONObject gespeichert. Abhängig vom Datentyp der Property wird der Wert als primitiver JSON-Typ oder in einem JSONArray oder JSONObject gespeichert. Das Format entspricht dabei der von der OX-JSON-API verwendeten Struktur.

Die Synchronisation in USM ist Folder-basiert. Jeder Folder wird individuell und unabhängig von anderen Foldern synchronisiert. Ein Folder kann dabei nur Elemente eines ContentTypes enthalten.

Die Folder-Hierarchie wird unabhängig von den Inhalten der einzelnen Folder synchronisiert. Dabei wird der gleiche Mechanismus wie für die Synchronisation eines Folders verwendet.

Es werden für die Synchronisation eines Folders oder der Folder-Hierarchie die gleichen USM-JSON-Befehle verwendet:

- Ist der Parameter folderid gesetzt, wird der Inhalt dieses Folders synchronisiert.
- Ist der Parameter folderid auf einem speziellen Wert gesetzt, wie „groups“ oder „resources“, werden entsprechend Groups und Resources synchronisiert.
- Ist der Parameter folderid nicht gesetzt, wird die Folder-Hierarchie synchronisiert.

Allgemein gilt bei der Übertragung von DataObjects, dass nicht übergebene Properties auf Ihrem Default-Wert oder ihrem vorher gesetzten Wert verbleiben. Wenn eine Property auf Ihren Default-Wert zurückgesetzt wird (oder werden soll), wird dies vom USM explizit gemeldet (bzw. muss vom Client explizit angegeben werden).

UUIDs

Für alle Befehle, die DataObjects als Parameter erwarten bzw. DataObjects in ihren Antworten liefern, gilt folgende allgemeine Regel:

- Jedes DataObject hat **immer** eine Property namens „uuid“, diese Property ist **immer** auf eine korrekte UUID gesetzt (bzw. muss immer gesetzt sein).

Dies gilt auch für Befehle, die es dem Client erlauben, nur bestimmte Properties von DataObjects abzufragen. Die „uuid“-Property wird immer mitgeliefert.

UUIDs werden in der JSON-API von USM immer als String in ihrer Normalform übergeben und erwartet. Beispiel:

```
"uuid": "550e8400-e29b-11d4-a716-446655440000"
```

ConflictResolution

Bei sich überschneidenden Änderungen von Client und Server können Konflikte auftreten. Konflikte sind dabei definiert als Änderungen sowohl vom Server als auch vom Client am gleichen DataObject seit der letzten Synchronisation, unabhängig davon, welche Properties jeweils geändert wurden. Das USM stellt verschiedene Konfliktlösungsstrategien zur Verfügung, die über den Parameter oder die Konfigurationseinstellung „conflictResolution“ eingestellt werden können:

1. „ERROR“ - Es wird eine Liste der sich überschneidenden Änderungen rückgemeldet
2. „USE_CLIENT“ - Die Server-Änderung wird rückgängig gemacht und die Client-Änderung übernommen
3. „USE_SERVER“ – Die Client-Änderung wird ignoriert und die Server-Änderung gemeldet

4. "ERROR_DELETE_OVER_CHANGE" - Wie ERROR, aber Löschungen haben eine höhere Priorität, d.h. eine gleichzeitige Änderung und Löschung führt nicht zum Fehler, sondern die Änderung wird stillschweigend überschrieben/ignoriert
5. "USE_CLIENT_DELETE_OVER_CHANGE" - Wie USE_CLIENT, aber Löschungen haben eine höhere Priorität
6. "USE_SERVER_DELETE_OVER_CHANGE" - Wie USE_SERVER, aber Löschungen haben eine höhere Priorität

Weitere Informationen zu den Konfliktlösungsstrategien sind auch hier zu finden:

<https://dev-wiki.open-xchange.com/cgi-bin/twiki/view/Main/ConflictResolution>

Attachments zu PIM-Objekten

Über die USM-JSON-API können auch Attachments zu PIM-Objekten synchronisiert werden. Dies wird über ein extra Feld bei allen PIM-Objekten realisiert ("attachments"). Das Feld wird nur gefüllt bei Änderungen an Attachments und für neue Objekte mit Attachments. Es enthält ein Array von Attachment JSONObjects. Jedes dieser JSONObjects enthält folgende Felder:

- uuid: Immer notwendig
- created_by (String)
- creation_date (Number)
- filename (String)
- file_size (Number)
- file_mimetype (String)
- rtf_flag (Boolean)
- data: base64-codierte Binärdaten des Attachments

Attachments können sich nach einer Erzeugung nicht ändern, d.h. bei einer Änderung muss das alte Attachment gelöscht und ein neues angelegt werden. Beim Anlegen von Attachments müssen alle Felder gesetzt sein (außer created_by und creation_date, die automatisch vom Server gesetzt werden). Falls bei einer Änderung bereits existierende Attachments an einem Objekt vorhanden sind, und diese beibehalten werden (sollen), wird für unveränderte Attachments nur das Feld uuid gesetzt. Falls ein Attachment entfernt wurde bzw. werden soll, wird es nicht mehr in dem übertragenen JSONArray aufgeführt.

Da Änderungen an Attachments nicht möglich sind (nur Löschungen und Erzeugungen), wird folgender vom normalen Sync-System unabhängiger Algorithmus als Konfliktlösung bei gleichzeitiger Änderungen der Attachments durch Client und Server benutzt:

- Bei gleichzeitigen Änderungen wird immer ein Merge der Client- und Server-Änderungen durchgeführt:
- Client-Löschungen werden auf dem Server durchgeführt, falls das Attachment nicht bereits auf dem Server ebenfalls gelöscht wurde
- Alle Client-Erzeugungen werden auf dem Server angelegt
- Alle Server-Erzeugungen werden an den Client gemeldet

Sync-Fehler-Rückgabe-Daten

Für den Fall, dass eine Synchronisation zwischen Client und Server aufgrund sich überschneidender Aktionen und der eingestellten ConflictResolution nicht möglich ist, liefern sowohl SyncInit als auch SyncUpdate eine spezielle Fehler-Antwort (StatusCode:9, „sync failed“) zurück, in der die sich überschneidenden Änderungen im Detail beschrieben sind.

Das Feld „errorDetails“ enthält ein JSONObject, in dem wiederum im Feld „conflicts“ ein JSONArray gespeichert ist. Jedes Array-Element entspricht einem Konflikt für eines der zu synchronisierenden DataObjects und ist wieder ein JSONObject mit 3 Feldern:

- „uuid“: Enthält die eindeutige UUID des DataObjects, für das ein Konflikt aufgetreten ist
- „client“: Enthält die Client-Änderung
- „server“: Enthält die Server-Änderung

Sowohl „client“ als auch „server“ enthalten als Wert entweder den String „deleted“, falls das Objekt gelöscht wurde, oder ein JSONObject, welches genau (und nur) die von der entsprechenden Stelle geänderten Felder des DataObjects enthält.

Beispiel einer Konflikt-Antwort:

Response :

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":9,
"errorDetails":{"conflicts":[{"uuid":"0123456789ABCDEF012345678
9ABCDEF",
"client":"deleted","server":{"field1":"value"}},{ "uuid":"001F2E
3D4C5B6A798897A6B5C4D3E2F1","client":{"field1":"value1"},"serve
r":{"field2":"value2"}}]}}
```

Fehlermeldungen für einzelne Elemente

Fehler, die eindeutig Client-Aktionen zu einzelnen Objekten zugeordnet werden können, werden von den Sync (SyncInit, SyncUpdate)-Befehlen im „errors“-Feld der Antwort zurückgemeldet. Hier wird für jeden aufgetretenen Fehler ein JSONObject unter der vom Client übergebenen UUID gemeldet. Das JSONObject wiederum enthält folgende Felder:

Name	Datentyp	Optional	Beschreibung
------	----------	----------	--------------

errorStatus	Integer	Nein	<p>Allgemeine Art des Fehlers (ein Subset der gleichen Codes wird auch vom MoveItems-Befehl verwendet):</p> <ol style="list-style-type: none"> 1. Destination does not exist * 2. Source does not exist * 3. Destination: no permission * 4. Source: no permission (all objects) * 5. Source: no permission (own objects) * 6. Other uncategorized error 7. Unknown UUID 8. No create permission 9. No change permission 10. No delete permission 11. PIM attachment creation failed 12. Parent folder not found 13. Mail Folder not found 14. Permission error from server 15. UID already exists on server 16. PIM attachment creation failed: max. number of attachments exceeded 17. PIM attachment creation failed: max. size (of all attachments of 1 object) exceeded 18. Multiple operations with same UUID <p>(*) Diese Codes werden nur vom MoveItems-Befehl verwendet</p>
errorCode	Integer	Nein	Spezifischer Fehlercode, der die Stelle im USM eindeutig spezifiziert, an der der Fehler aufgetreten ist
data	JSONObject	Ja	Komplettes DataObject mit allen gesetzten Feldern. Wird nicht gemeldet, wenn eine unbekannte UUID vom Client übergeben wurde. Im Falle eines Create werden die vom Client übergebenen Daten gemeldet, sonst die zuletzt synchronisierten Daten
errorDetails	JSONObject	Ja	Weiterführende Informationen zum Fehler, soweit vorhanden
oxError	JSONObject	Ja	Im Falle eines Fehlers, der vom OX-Server gemeldet wurde: Alle vom OX-Server gemeldeten Fehlerinformationen

Der Fehler-Status 11 (PIM attachment creation failed) wird vom USM dann geliefert, wenn das reine Anlegen eines Attachments vom OX-Server mit einem Fehler quittiert wurde. Dies ist normalerweise darauf zurückzuführen, dass eine Beschränkung im OX-Server (z.B. Größen-Limit für Anhänge) das Speichern verhindert. Der genaue vom OX-Server gemeldete Fehler ist dann im Feld „oxError“ gespeichert. Dieser Fehler-Status behindert nicht die

restliche Synchronisierung des Objekts, d.h. es wird weiter synchronisiert, als hätte der Client das Attachment nicht übergeben. Da protokollbedingt nur 1 Fehler pro Element gemeldet werden kann, wird nur der 1. Fehler beim Anlegen mehrerer Attachments gemeldet. Sollten in der Nachbearbeitung zu dem Objekt noch weitere Fehler auftreten (falls z.B. ein vom Client initiiertes Löschen eines anderen Attachments aufgrund eines Server-Fehlers nicht möglich ist), so haben diese eine höhere Priorität und überschreiben den Fehler-Status 11.

Falls der Client mehrere PIM Attachments zu einem Objekt in einem Synchronisationsvorgang anlegen wollte oder falls bei einem Objekt, zu dem ein Attachment angelegt werden sollte, ein anderer Fehler gemeldet wird, sollte der Client beim nächsten Synchronisationsvorgang mittels des „refresh“-Parameters den aktuellen Serverzustand des entsprechenden Objekts abfragen, um einen synchronen Zustand sicherzustellen.

Der Fehler-Status 12 (Parent folder not found) wird vom USM gemeldet, wenn beim Anlegen eines Folders der Parent-Folder nicht mehr verfügbar ist (z.B. weil der Parent-Folder mittlerweile gelöscht wurde, oder weil die Zugriffsberechtigungen so geändert wurden, dass der Folder für den aktuellen Benutzer nicht mehr sichtbar ist). Der genaue vom OX-Server gemeldete Fehler ist unter „oxError“ aufgeführt.

Der Fehler-Status 18 (Multiple operations with same UUID) wird gemeldet, wenn die gleiche UUID für verschiedene Operationen verwendet wurde, z.B. mehrere Objekte mit gleicher UUID erzeugt, oder mehrere konkurrierende Client-Operationen für die gleiche UUID ausgeführt). Alle Operationen für diese Objekte schlagen fehl, es wird nur 1 Fehler für die UUID wird gemeldet.

Sync Init

Pfad:

<Basis-URL>/syncInit

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
folderid	String	Ja		UUID des Folders od. „groups“ od. „resources“
created	JSONArray	Ja		Liste von DataObjects
limit	Integer	Ja	∞	maximale Anzahl der Änderungen, die zurückgeliefert werden sollen
conflictResolution	String	Ja	Config	Zu verwendende ConflictResolution

Response:

JSONObject

Name	Datentyp	Optional	Bedeutung
syncid	Long	Nein	Sync-ID für folgende SyncUpdate-Befehle
moreAvailable	Boolean	Ja	„true“ falls noch weitere Server-Änderungen vorhanden sind

created	JSONArray	Ja	Liste von neuen Objekten auf dem Server
modified	JSONArray	Ja	Liste von geänderten Objekten auf dem Server
errors	JSONObject	Ja	Rückmeldung von Fehlern beim Ausführen von Server-Updates

Beschreibung:

Dieser Befehl führt die initiale Synchronisation der Folder-Hierarchie („folderid“ nicht gesetzt) oder eines Folders („folderid“ gesetzt) durch. Falls der „folderid“ einer von den speziellen Werten „groups“ oder „resources“ enthält, werden entsprechend die Groups oder die Resources synchronisiert. Der Client sendet hierzu in „created“ eine Liste aller aktuell auf dem Client vorhandenen DataObjects (falls auf dem Client keine Objekte vorhanden sind, darf der Parameter „created“ entfallen). Es müssen dabei nur die Properties eines DataObjects gesetzt sein, die nicht auf dem Default-Wert sind. Optional kann der Client auch die zu verwendende ConflictResolution übergeben, wird keine spezifiziert, wird die in der Konfiguration eingetragene verwendet. Weiterhin kann der Client ein Limit spezifizieren, welches die maximale Anzahl der durchzuführenden Änderungen angibt, die vom USM gemeldet werden sollen.

Das USM liefert als Ergebnis eine Sync-ID, die für zukünftige Sync-Update-Befehle verwendet werden kann, sowie eine Liste von neu zu erzeugenden und zu ändernden DataObjects. Falls die Anzahl der gemeldeten DataObjects die spezifizierte Obergrenze überschreitet, wird zusätzlich das Flag „moreAvailable“ gesetzt, um den Client zu informieren, dass weitere Server-Änderungen (mittels eines SyncUpdate-Befehls und der von USM gelieferten Sync-ID) abgerufen werden können.

Für Erzeugungen setzt das USM nur die Properties in den Objekten, die nicht auf dem Default-Wert sind. Für Änderungen werden nur die Properties gemeldet, die sich geändert haben. Nicht gemeldete Properties dürfen also **nicht** gelöscht bzw. auf leer gesetzt werden.

Falls bei der Synchronisation Änderung auf dem OX-Server durchgeführt werden müssen und beim Ausführen der entsprechenden Befehle Fehler auftreten (weil z.B. ungültige Werte oder Wert-Kombinationen gesetzt werden oder weil keine Schreib-Berechtigung vorhanden ist), liefert USM im Rückgabe-Feld „errors“ für alle nicht durchgeführten Aktionen die entsprechende Fehlermeldung zurück. „errors“ enthält dabei ein JSONObject. In diesem sind als Keys die UUIDs der Objekte aufgeführt, für die Fehler auftraten, als Value dazu wird ein JSON-Objekt übergeben, in dem die Fehler-Codes und -Details (analog zum allgemeinen Fehlerfall beim Ausführen eines USM-JSON-API-Befehls) aufgelistet sind.

Beispiel:

Request:

```
PUT http://localhost/usm/syncInit
Content-Type: text/javascript

{"sessionid":"USM-1","created":[],"limit":10}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1,"data":{"syncid":1,"moreAvailable":"true",
"created":[{"field":"value",...},{ "field":"value",...},...]}}
```

Sync Update

Pfad:

<Basis-URL>/syncUpdate

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
folderid	String	Ja		UUID des Folders od. „groups“ od. „resources“
syncid	Long	Nein		Sync-ID des letzten SyncInit/SyncUpdate
created	JSONArray	Ja		Liste von neuen DataObjects
modified	JSONArray	Ja		Liste von geänderten DataObjects
deleted	JSONArray	Ja		Liste von gelöschten DataObject- UUIDs
limit	Integer	Ja	∞	maximale Anzahl der Änderungen, die zurückgeliefert werden sollen
conflictResolution	String	Ja	Config	Zu verwendende ConflictResolution
refresh	JSONArray	Ja		Liste von UUIDs, für deren Objekte der komplette aktuelle Inhalt übermittelt werden soll

Response:

JSONObject

Name	Datentyp	Optional	Bedeutung
syncid	Long	Nein	Sync-ID für folgende SyncUpdate-Befehle
moreAvailable	Boolean	Ja	„true“ falls noch weitere Server-Änderungen vorhanden sind
created	JSONArray	Ja	Liste von neuen Objekten auf dem Server
modified	JSONArray	Ja	Liste von geänderten Objekten auf dem Server
deleted	JSONArray	Ja	Liste von gelöschten DataObject-UUIDs
errors	JSONObject	Ja	Rückmeldung von Fehlern beim Ausführen von Server-Updates

Beschreibung:

Dieser Befehl führt eine aktualisierende Synchronisation der Folder-Hierarchie („folderid“ nicht gesetzt) oder eines Folders („folderid“ gesetzt) durch. Falls der „folderid“ einer von den speziellen Werten „groups“ oder „resources“ enthält, werden entsprechend die Groups oder die Resources synchronisiert. Der Client sendet hierzu alle auf dem Client seit der letzten Synchronisation (gemeldet über „syncid“) durchgeführten Erzeugungen (in „created“), Änderungen (in „modified“) und Löschungen (Liste von UUIDs in „deleted“) von DataObjects. Im Falle von Synchronisation von Groups oder Resources darf der Client keine Änderungen senden.

Desweiteren kann der Client auch die zu verwendende ConflictResolution übergeben, sowie ein Limit spezifizieren, welches die maximale Anzahl der durchzuführenden Änderungen angibt, die vom USM in der Antwort gemeldet werden sollen.

Das USM liefert als Ergebnis eine Sync-ID, die für weitere Sync-Update-Befehle verwendet werden kann, sowie eine Liste von neu zu erzeugenden, zu ändernden und zu löschenden DataObjects. Falls die Anzahl der gemeldeten DataObjects die spezifizierte Obergrenze überschreitet, wird zusätzlich das Flag „moreAvailable“ gesetzt, um den Client zu informieren, dass weitere Server-Änderungen (mittels eines SyncUpdate-Befehls und der von USM gelieferten Sync-ID) abgerufen werden können.

Für Erzeugungen setzt das USM nur die Properties in den Objekten, die nicht auf dem Default-Wert sind. Für Änderungen werden nur die Properties gemeldet, die sich geändert haben. Nicht gemeldete Properties dürfen also nicht gelöscht bzw. auf leer gesetzt werden. Für Löschungen werden vom USM nur die UUIDs der entsprechenden DataObjects gemeldet. Analog muss der Client bei Änderungen nur geänderte Properties melden, soll eine Property auf den Default-Wert zurückgesetzt werden, muss dies vom Client explizit durchgeführt werden. Vom Client nicht übergebene Properties werden bei einer Änderung an einem DataObject nicht modifiziert.

Falls bei der Synchronisation Änderung auf dem OX-Server durchgeführt werden müssen und beim Ausführen der entsprechenden Befehle Fehler auftreten (weil z.B. ungültige Werte oder Wert-Kombinationen gesetzt werden oder weil keine Schreib-Berechtigung vorhanden ist), liefert USM im Rückgabe-Feld „errors“ für alle nicht durchgeführten Aktionen die entsprechende Fehlermeldung zurück. „errors“ enthält dabei ein JSONObject. In diesem sind als Keys die UUIDs der Objekte aufgeführt, für die Fehler auftraten, als Value dazu wird ein JSON-Objekt übergeben, in dem die Fehler-Codes und -Details (analog zum allgemeinen Fehlerfall beim Ausführen eines USM-JSON-API-Befehls) aufgelistet sind.

Falls eine Synchronisation auf dem Client nicht vollständig abgeschlossen werden konnte, und es dem Client nicht möglich ist, alle Änderungen des Synchronisationsvorgangs rückgängig zu machen, befinden sich einige Objekte auf dem Client evtl. in einem inkonsistenten Zustand. Um hier die Notwendigkeit eines Neustarts der Synchronisation mittels „SyncInit“ zu umgehen, wurde der Parameter „refresh“ eingeführt. Über diesen optionalen Parameter kann der Client eine Liste von UUIDs übergeben. Der Server wird für diese UUIDs immer eine Antwort zurückliefern, die den kompletten Zustand des Objekts auf Serverseite widerspiegelt. Für auf dem Server existierende Objekte wird das komplette Objekt mit allen Feld-Inhalten (inkl. der Felder auf Default-Wert) als „created“ gemeldet (unabhängig davon, ob das Objekt im Ausgangs-Sync-State bereits vorhanden war). Ist ein Objekt mit der angegebenen UUID auf dem Server in diesem Folder nicht (oder nicht mehr) vorhanden, wird die UUID unter „deleted“ zurückgemeldet. Dies gilt auch für UUIDs, die dem Server komplett unbekannt sind. Falls vom Client eine syntaktisch inkorrekte UUID übergeben wurde, wird eine entsprechende Fehlermeldung (Status 7) unter „errors“ gemeldet. Falls zu dem Objekt erweiterte Daten vom OX-Server gelesen werden müssen (z.B. Mail-Struktur, PIM-Attachments, Kontakt-Bild) und dabei ein Fehler auftrat, wird dies ebenfalls unter „errors“ mit einer entsprechenden Fehlermeldung zurückgemeldet.

Achtung: Die Resulte zu den in „refresh“ übergebenen Objekten werden vom Server auf jeden Fall zurückgeliefert, unabhängig von einem vom Client spezifizierten „limit“-Parameter und zusätzlich zu den normalen limitierten Server-Änderungen. Falls ein „refresh“-Parameter übergeben wurde, werden also Server-Objekte zwischen dem Wert von „limit“ und diesem Wert plus der Anzahl der in „refresh“ übergebenen UUIDs zurückgeliefert.

Warnung: Da die in „refresh“ übergebenen Objekte auf jeden Fall zurückgeliefert werden, sollte ein Client darauf achten, nicht zu viele UUIDs in diesem Parameter zu übergeben. Andernfalls kann die Abarbeitung des Requests sehr lange dauern und im schlimmsten Falle zum Abbruch der Verbindung führen. Auch die Datenmenge kann bei entsprechenden Objekten (z.B. Email) sehr umfangreich werden. Im Zweifelsfall sollte der Client mehrere syncUpdates mit „refresh“-Parameter nacheinander durchführen, um alle inkonsistenten Objekte zu erhalten, oder bei vielen Objekten direkt über die Ausführung von „SyncInit“ die Synchronisierung neu initiieren.

Beispiel:

Request:

```
PUT http://localhost/usm/syncUpdate
Content-Type: text/javascript

{"sessionid":"USM-1","syncid":1,"limit":10}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1,"data":{"syncid":2,"created":[{"field":"value",...}, {"field":"value",...},...]}}
```

Sync Estimate

Pfad:

<Basis-URL>/syncEstimate

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
folderid	String	Ja		UUID des Folders od. „groups“ od. „resources“
syncid	Long	Ja		Sync-ID des letzten SyncInit/SyncUpdate

Response:

JSONObject

Name	Datentyp	Optional	Bedeutung
created	Integer	Nein	Anzahl der neuen DataObjects

modified	Integer	Nein	Anzahl der geänderten DataObjects
deleted	Integer	Nein	Anzahl der gelöschten DataObjects

Beschreibung:

Dieser Befehl liefert eine Abschätzung der auf dem Server vorhandenen Erzeugungen, Änderungen und Löschungen für die Folder-Hierarchie („folderid“ nicht gesetzt) oder einen Folder („folderid“ gesetzt). Falls der „folderid“ einer von den speziellen Werten „groups“ oder „resources“ enthält, werden entsprechend die Änderungen der Groups oder der Resources abgeschätzt.

Wird vom Client der Parameter „syncid“ gesetzt, so wird die Abschätzung auf Basis des übergebenen Sync-Zustands durchgeführt. Wird der Parameter nicht gesetzt, wird die Abschätzung für einen initialen Sync-Vorgang durchgeführt (in diesem Fall ist der Rückgabewert von „modified“ und „deleted“ immer 0).

Dieser Befehl kann vom Client z.B. dazu benutzt werden, einen Progress- oder Bestätigungs-Dialog für die bzw. vor der Synchronisation darzustellen. Folgendes ist dabei zu beachten:

- Die gelieferten Werte stimmen nicht immer genau mit der tatsächlich vorhandenen Anzahl an Änderungen überein. Eine Rückmeldung von 0 Änderungen stellt insbesondere keine Sicherheit dafür da, dass keine Änderungen vorhanden sind.
- Die Bestimmung der Anzahl an Änderungen ist auf der Serverseite ähnlich aufwendig wie der eigentliche Synchronisationsvorgang. Der Befehl sollte also mit Bedacht eingesetzt werden (insbesondere sollte bei aufgeteilten Synchronisationsvorgängen, d.h. „limit“ und „moreAvailable“ gesetzt, nicht nach jedem SyncUpdate-Befehl neu ein SyncEstimate aufgerufen werden)

Beispiel:

Request:

```
PUT http://localhost/usm/syncEstimate
Content-Type: text/javascript

{"sessionId":"USM-1","syncid":1}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1,"data":{"created":2,"modified":0,"deleted":1}}
```

Set Push Notification

Achtung: Der Server sendet zurzeit keine Push-Benachrichtigungen. Der Client kann den Server-Zustand mit dem Befehl „Ping“ abfragen. Der Befehl setPushNotification ist deshalb zur Zeit deaktiviert und nicht zugreifbar, kann aber evtl. in einer zukünftigen Version wieder aktiviert werden.

Pfad:

<Basis-URL>/setPushNotification

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
type	String	Nein		Art der gewünschten Benachrichtigung
host	String	Ja		Host, der benachrichtigt werden soll
port	Integer	Ja		Port, an den Benachrichtigung geschickt werden soll

Response:

Leer

Beschreibung:

Mit diesem Befehl setzt der Client die gewünschte Methode für Push-Benachrichtigungen. Eine Push-Benachrichtigung wird gesendet, wenn eine Änderung auf dem Server auftritt. Die Art der Benachrichtigung wird über den „type“-Parameter festgelegt:

- „none“: Die Push-Benachrichtigungen sind abgeschaltet. Es muss kein weiterer Parameter übergeben werden.
- „udp“: Push-Benachrichtigungen sollen über UDP durchgeführt werden. Über die notwendigen Parameter „host“ und „port“ spezifiziert der Client die Details der Benachrichtigung.

In zukünftigen Versionen ist es denkbar, dass alternative Push-Benachrichtigungssysteme implementiert werden. Diese werden über andere Werte für den „type“-Parameter bestimmt und über eigene zusätzliche Parameter konfiguriert.

Beispiel:

Request:

```
PUT http://localhost/usm/setPushNotification
Content-Type: text/javascript

{"sessionid":"USM-1","type":"udp",
"host":"127.0.0.1","port":"12345"}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1}
```

Get Cached Data

Pfad:

<Basis-URL>/getCachedData

Parameter:

Name	Datentyp	Optional	Default	Beschreibung
------	----------	----------	---------	--------------

			Wert	
sessionid	String	Nein		
syncid	Long	Nein		
folderid	String	Ja		UUID des Folders

Response:

JSONArray von DataObjects

Beschreibung:

Mit diesem Befehl kann der Client für einen gegebenen Sync-Zustand die im USM hinterlegten DataObjects mit ihren aktuellen Werten abfragen. Dieser Befehl sollte im Allgemeinen für Clients nicht relevant sein. Achtung: Es werden alle gespeicherten Elemente zurückgeliefert, die Datenmenge kann entsprechend hoch sein.

Beispiel:

Request:

```
PUT http://localhost/usm/getCachedData
Content-Type: text/javascript

{"sessionid":"USM-1","syncid":1}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1,"data":[{"field":"value",...}, {"field":"value",...}
,...]}
```

Read Data

Pfad:

<Basis-URL>/readData

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
syncid	Long	Nein		
folderid	String	Nein		UUID des Folders
objectids	JSONArray	Nein		Liste von DataObject-UUIDs

properties	JSONArray	Nein		Liste von Property-Namen
------------	-----------	------	--	--------------------------

Response:

JSONArray von DataObjects

Beschreibung:

Mit diesem Befehl kann der Client für bestimmte DataObjects in einem Folder den aktuell auf dem Server gespeicherten Inhalt bestimmter Properties abfragen. Dies ist z.B. dann interessant, wenn sich bestimmte Properties z.B. nach dem Anlegen nie ändern können (etwa Body oder Attachments einer Email) und damit unnötig Platz im Synchronisations-Cache beanspruchen würden. Das USM liefert ein JSONArray von DataObjects zurück, in denen genau (und nur) die angefragten Properties gesetzt sind.

Beispiel:

Request:

```
PUT http://localhost/usm/readData
Content-Type: text/javascript

{"sessionid":"USM-1","syncid":1, "folderid":"id1",
"objectids":["id1"],"properties":["field"]}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1,"data":[{"field":"value"}]}
```

Send Mail

Pfad:

<Basis-URL>/sendMail

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
mail	JSONObject	Nein		Zu versendende Email

Response:

Leer

Beschreibung:

Dieser Befehl dient dazu, eine Email zu versenden. Der Client spezifiziert die Email über die entsprechenden Properties im übergebenen DataObject. Das USM generiert aus diesen Daten eine entsprechende (MIME-)Email und versendet sie über den OX-Server.

Beispiel:

Request:

```
PUT http://localhost/usm/sendMail
Content-Type: text/javascript

{"sessionId":"USM-1","mail":{"headers":{...},...}}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1}
```

Resolve Recipients

Pfad:

<Basis-URL>/resolveRecipients

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionId	String	Nein		
properties	JSONArray	Nein		Liste von abzufragenden Properties der „contact“-DataObjects
names	JSONArray	Nein		Liste von Such-Strings

Response:

JSONArray von DataObjects

Beschreibung:

Dieser Befehl dient dazu, alle Kontakt-Folders nach Kontakten mit bestimmten Namen zu durchsuchen. Die Properties „last_name“, „first_name“ und „display_name“ aller Kontakte werden mit der übergebenen Namensliste verglichen. Das USM liefert ein JSONArray von DataObjects zurück, in denen genau (und nur) die angefragten Properties gesetzt sind und bei denen eine potentielle Übereinstimmung festgestellt wurde.

Beispiel:

Request:

```
PUT http://localhost/usm/resolveRecipients
Content-Type: text/javascript
```

```
{"sessionId":"USM-1","properties":["field"],"names":["user"]}
```

Response:

HTTP/1.1 200 OK

Content-Type: text/javascript

```
{"status":1,"data":[{"field":"value1"},{"field":"value2"},...]}
```

Move Items

Pfad:

<Basis-URL>/moveItems

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionId	Long	Nein		
sourcefolderid	String	Nein		UUID des Source-Folders
sourcesyncid	Long	Ja		Sync-ID des letzten Sync-Vorgangs des Quell-Folders
destinationfolderid	String	Nein		UUID des Ziel-Folders
destinationsyncid	Long	Ja		Sync-ID des letzten Sync-Vorgangs des Ziel-Folders
objectids	JSONArray	Nein		Liste von UUIDs der DataObjects die verschoben werden sollen

Response:

JSONObject

Name	Datentyp	Optional	Bedeutung
sourcesyncid	Long	Ja	neue Sync-ID für den Quell-Folder
destinationsyncid	Long	Ja	neue Sync-ID für den Ziel-Folder
errors	JSONObject	Ja	JSON Objekt - enthält die UUID des Objektes das nicht verschoben werden konnte als Key und ein JSON Objekt als value, das aus ein Fehler-StatusCode und die Fehler Details besteht.

Beschreibung:

Mit diesem Befehl kann der Client DataObjects von einem Folder in einen anderen (kompatiblen) Folder verschieben. Ausgehend vom Sync-Zustand in Quellordner werden alle in „objectids“ übergebenen DataObjects vom USM in den Ziel-Folder verschoben. Für Email Ordner wird der neueste Sync-Zustand des Ziel-Ordners geändert, und eine neue „destinationsyncid“ für diesen Ordner geliefert. Für alle anderen Ordner-Typen ist dieser Vorgang nicht notwendig. Die Parameter „sourcesyncid“ und „destinationsyncid“ sind Optional. Falls bekommen, in der Regel liefert das USM die gleichen syncids zurück, außer die destinationsyncid für Mail-Ordner die immer geliefert wird.

Dieser Befehl dient vor allem dazu, dass Verschiebungen von großen DataObjects (z.B. Emails) nicht als Löschungen und Neu-Erzeugungen gemeldet werden, was zu erhöhtem Transfervolumen zwischen Client und Server führen würde.

Falls beim Ausführen der entsprechenden Befehle zum Verschieben Fehler auftreten, liefert USM im Rückgabe-Feld „errors“ für alle nicht durchgeführten Aktionen die entsprechende Fehlermeldung zurück. „errors“ enthält dabei ein JSONObject. In diesem sind als Keys die UUIDs der Objekte aufgeführt, für die Fehler auftraten, als Value dazu wird ein JSON-Objekt übergeben, in dem die Fehler-StatusCodes und -Details aufgelistet sind.

Folgende Fehler-StatusCodes werden geliefert:

- Ziel-Ordner existiert nicht = 1
- Quell-Ordner existiert nicht = 2
- Keine Berechtigung in dem Ziel-Ordner Objekte zu speichern = 3
- Keine Berechtigung in dem Quell-Ordner fremde Objekte zu löschen = 4
- Keine Berechtigung in dem Quell-Ordner eigene Objekte zu löschen = 5
- Andere Fehler = 6
- Unbekannte UUID = 7

Beispiel:

Request:

```
PUT http://localhost/usm/moveItems
Content-Type: text/javascript

{"sessionid":"USM-1","sourcefolderid":
"0102030405060708090A0B0C0D0F00","sourcesyncid":1,
"destinationfolderid":"0102030405060708090A0B0C0D0F01",
"destinationsyncid":4,"objectids":["0102030405060708090A0B0C0D0
F02", "0102030405060708090A0B0C0D0F03"]}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1,"data":{"sourcesyncid":2,"destinationsyncid":5,"err
ors":{"0102030405060708090A0B0C0D0F02":{"...}}}}
```

Ping

Pfad:

<Basis-URL>/ping

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	Long	Nein		
objectids	JSONArray	Ja		Liste von UUIDs der Folders deren Zustand zurückgemeldet werden soll. Enthält auch die Context-UUID wenn Änderungen in der Folder-Struktur zurückgemeldet werden sollen

Response:

JSONObject

Name	Datentyp	Optional	Bedeutung
interval	Long	Nein	zu welchem Intervall (in ms) soll der Client die nächste Ping Requests senden
objectids	JSONArray	Ja	Liste von UUIDs der Folders die auf dem Server Änderungen enthalten. Enthält auch die Context-UUID wenn vom Server Änderungen in der Folder-Struktur gemeldet wurden.

Beschreibung:

Mit diesem Befehl kann der Client abfragen ob seit dem letzten Sync Änderungen vom OX-Server gemeldet wurden. Optional kann der Client eine Liste von Folder-UUIDs, die von Interesse sind, übergeben. Diese Liste kann die Context-UUID enthalten, wenn Änderungen in der Folder-Struktur zurückgemeldet werden sollen. Wenn keine Liste übergeben wurde, dann meldet der Server Änderungen in allen Folders und in der Folder-Struktur.

Änderungen in Groups und Resources werden nicht gemeldet.

Falls der OX-Server keine Änderungen seit dem letzten Sync gemeldet hat, wird der „objectids“ Parameter in der Antwort nicht gesetzt.

Beispiel:

Request:

```
PUT http://localhost/usm/ping
Content-Type: text/javascript

{"sessionid":"USM-1",
"objectids":["0102030405060708090A0B0C0D0F02",
"0102030405060708090A0B0C0D0F03"]}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript
```

```
{ "status":1, "data":{"interval":5000,
"objectids":["0102030405060708090A0B0C0D0F02"]}}
```

Resolve OX IDs

Pfad:

<Basis-URL>/resolveIDs

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
user	String	Nein		
device	String	Nein		
password	String	Nein		
folder_uuid	String	Ja		UUID des Parent Folders. Falls nicht gesetzt, muss eine liste von folder uuids in dem Parameter objects_uuids übergeben werden
object_uuids	JSONArray	Nein		Liste von UUIDs der Objekte deren OX-IDs zurückgemeldet werden sollen. Enthält folder uuids falls der Parameter folder_uuid nicht gesetzt ist.

Response:

JSONObject

Name	Datentyp	Optional	Bedeutung
folder_id	String	Ja	Die OX ID des Folders dessen uuid der Client im Parameter folder_uuid gesendet hat
object_ids	JSONObject	Nein	Enthält die vom Client gesendeten object uuids als keys und die jeweils passende ox id als value

Beschreibung:

Mit diesem Befehl kann der Client die OX IDs für eine Liste von uuids abfragen.

Beispiel:

Request:

```
PUT http://localhost/usm/resolveIDs
Content-Type: text/javascript
```

```
{
  "user": "user1",
  "device": "android",
  "password": "secret",
  "folder_uuid": "0102030405060708090A0B0C0D0F02",
  "object_uuids": [
    "0102030405060708090A0B0C0D0F05",
    "0102030405060708090A0B0C0D0F03"
  ]
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript
```

```
{
  "status": 1,
  "data": {
    "folder_id": "30",
    "object_ids": {
      "0102030405060708090A0B0C0D0F05": "1",
      "0102030405060708090A0B0C0D0F03": "2"
    }
  }
}
```

Invalidate Objects

Pfad:

<Basis-URL>/invalidateObjects

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
syncid	String	Nein		
folderid	String	Ja		UUID des Parent Folders. Falls nicht gesetzt, muss eine liste von folder uuids in dem Parameter objectids übergeben werden
objectids	JSONArray	Nein		Liste von UUIDs der Objekte die aus dem sync state entfernt werden sollen. Enthält folder uuids falls der Parameter folderid nicht gesetzt ist.

Response:

JSONObject

Name	Datentyp	Optional	Bedeutung
syncid	String	Nein	Die neue syncid
objects	JSONObject	Nein	Enthält die vom Client gesendeten object uuids als keys und status „ok“ oder „not found“

Beschreibung:

Mit diesem Befehl kann der Client einzelne Objekte als invalide markieren, d.h. als nicht korrekt vom Client empfangen oder aus sonstigem Grunde nicht korrekt synchronisiert. Der Befehl entfernt diese Objekte aus dem übergebenen Sync-Zustand und liefert die syncid eines entsprechend modifizierten Sync-Zustands zurück. Beim nächsten SyncUpdate-Befehl

mit der neuen syncid werden diese Objekte neu als Erzeugungen vom Server gemeldet, falls Sie auf dem Server noch vorhanden sind, d.h. nicht zwischen-zeitlich durch andere Clients/Nutzer gelöscht wurden.

Beispiel:

Request:

```
PUT http://localhost/usm/invalidateObjects
Content-Type: text/javascript

{"sessionId":"USM-1","folderid":
"0102030405060708090A0B0C0D0F00","syncid":1,
"objectids":["0102030405060708090A0B0C0D0F02",
"0102030405060708090A0B0C0D0F03"]}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1, "data":{"syncid ":"2",
"objects":{"0102030405060708090A0B0C0D0F02":"ok",
"0102030405060708090A0B0C0D0F03": „ok“}}}
```

Locate Objects

Pfad:

<Basis-URL>/locateObjects

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionId	String	Nein		
objectids	JSONArray	Nein		Liste von UUIDs der Objekte, für die der Parent-Folder ermittelt werden soll

Response:

JSONObject

Name	Datentyp	Optional	Bedeutung
<object-uuid>	String	Nein	UUID des Parent-Folders, oder falls Objekt selbst ein Folder war, der Text „folder“. Falls das Objekte nicht lokalisiert werden konnte, der Text „not found“

Beschreibung:

Mit diesem Befehl kann der Client für einzelne Objekte feststellen, in welchem Folder sie sich befinden. Für Objekte(Mail, Kontakt, Termin,...) in einem Folder wird die UUID des enthaltenden Folder zurück geliefert. Für Folder selbst wird der Text „folder“ in der Antwort geliefert. Konnte eine übergebene UUID in keinem aktuellen Sync-State lokalisiert werden, wird für diese UUID der Wert „not found“ geliefert.

Es ist möglich, dass ein Objekt in Sync-States zu verschiedenen Foldern gefunden wird, wenn das Objekt auf dem Server verschoben wurde, und seit der Verschiebung nur im Zielfolder eine Synchronisierung durchgeführt wurde. Das USM meldet bei mehrfachem Vorkommen den Folder, mit der aktuellsten Sync-ID.

Beispiel:

Request:

```
PUT http://localhost/usm/locateObjects
Content-Type: text/javascript

{"sessionid":"USM-1",
"objectids":["0102030405060708090A0B0C0D0F02",
"0102030405060708090A0B0C0D0F03",
"0102030405060708090A0B0C0D0F04"]}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1, "data":["0102030405060708090A0B0C0D0F02":
"0102030405060708090A0B0C0D0F0F",
"0102030405060708090A0B0C0D0F03": "folder",
"0102030405060708090A0B0C0D0F04": "not found"]}
```

Set Confirmation Status

Pfad:

<Basis-URL>/setConfirmationStatus

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
uid	String	Nein		UID (nicht UUID) des Appointments/Tasks, dessen Confirmation Status geändert werden soll.
confirmation	Integer	Nein		0 = none 1 = accepted 2 = declined 3 = tentative
confirmmessage	String	Nein		Die Confirm Message des Users

Response:

Leer

Beschreibung:

Da es vorkommen kann, daß ein User Teilnehmer eines Appointments/Tasks ist, für welches er keine Schreibberechtigung besitzt, ist es nicht immer möglich, den Confirmation Status des Users über einen normalen SyncUpdate zu ändern.

Mit dem Befehl `setConfirmationStatus` kann der Client den Confirmation Status (keiner/akzeptiert/abgelehnt/vorläufig) außerhalb eines normalen Syncs für den aktuell angemeldeten User setzen. Dieser Befehl kann für Appointments und für Tasks verwendet werden.

Wenn der Befehl erfolgreich ausgeführt wird, dann wird eine leere Response mit dem einzigen Feld „status“ zurückgegeben. Im Fehlerfall wird ein Error JSONObject analog zu den anderen USM Befehlen zurückgegeben.

Der Befehl wird außerhalb des normalen Synchronisations-Mechanismus ausgeführt, d.h. die Änderung wird dem Client im nächsten SyncUpdate des entsprechenden Folders mitgeteilt.

Beispiel:

Request:

```
PUT http://localhost/usm/setConfirmationStatus
Content-Type: text/javascript

{ "sessionId": "3cc36088-bc1f-4047-bd9f-701057cd1496",
  "uid": "dac376af-3292-451c-aa82-7f7ce8e6cf8d",
  "confirmation": 1,
  "confirmmessage": "Please provide an LCD projector" }
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1}
```

Erweiterungen USM-JSON-API (Version 2.0.0)

Mit der Versionsnummer 2.0.0 der USM-JSON-API werden Erweiterungen eingeführt, die eine verbesserte Handhabung von großen Attachments zur Verfügung stellen, und auch die Nutzung von sehr großen Attachments ermöglichen. Dies betrifft sowohl PIM-Attachments (also zu einzelnen PIM-Objekten hinzugefügte Dateien) als auch Email-Attachments (in der MIME-Mail-Struktur vorhandene große Datenobjekte).

In früheren Versionen der USM-JSON-API wurden Attachments immer inline in den einzelnen Befehlen (`syncInit`, `syncUpdate`, `sendMail`) mit übertragen. In Version 2.0.0 ist es möglich, den Inhalt eines Attachments durch eine Referenz zu ersetzen. Diese wird über das Feld „tempid“ identifiziert, welches immer einen String enthält. Über zusätzliche Befehle ist es dem Client möglich, den referenzierten Inhalt zu übertragen. Es gilt hier 2 grundsätzliche Fälle zu unterscheiden:

1. Übertragung von Attachments vom Server zum Client

- Bei einer Synchronisation (syncInit, syncUpdate) meldet der Server statt eines „data“-Feldes ein „tempid“-Feld, welches eine Referenz auf die Daten darstellt.
- Der Client lädt nach der Synchronisation den Inhalt des Attachments (d.h. die in früheren Versionen in „data“ enthaltenen Daten) über den neuen Befehl „downloadData“.
- Für kleine Attachments kann der Server weiterhin direkt das „data“-Feld setzen. Die Grenze, ab wann „data“ durch eine „tempid“ ersetzt wird, hängt von der Server-Konfiguration ab (default: 1 MB).

2. Übertragung von Attachments vom Client zum Server

- Der Client sendet vor einer Synchronisation mittels des neuen Befehls „uploadData“ Daten unter einer von ihm definierten „tempid“ zum Server.
- Bei einer Synchronisation (syncInit, syncUpdate) oder beim Senden einer Email (sendMail) meldet der Client statt des „data“-Feldes ein „tempid“-Feld, welches die von ihm vorher genutzte „tempid“ enthält. Der Server nutzt für die Inhalte an dieser Stelle die vorher vom Client gesendeten Daten.
- Der Client entscheidet, für welche Attachments er die Daten direkt über „data“ oder indirekt über „tempid“ zum Server überträgt. Um Verarbeitungsprobleme auf dem Server zu vermeiden, muss er sehr große Attachments über eine Referenzierung mittels „tempid“ übertragen. Auch bei einer sehr großen Anzahl an Attachment-Änderungen muss er sicherstellen, dass die Menge der in einem syncInit/syncUpdate/sendMail übertragenen Daten nicht zu groß wird.

Beide Fälle können für eine Synchronisation mehrfach (d.h. für mehrere Attachments mit unterschiedlichen „tempid“-Werten) und auch gemischt (z.B. Synchronisation mit Client- und Server-Änderungen an Attachments) auftreten.

Anwendungsmöglichkeiten der Ersetzung mittels „tempid“

Der Server kann an folgenden Stellen „tempid“-Felder generieren:

- PIM-Attachments: Innerhalb eines Objekts der im Feld „attachments“ gelieferten Liste von Attachment-Objekten, statt des Feldes „data“
- Emails: Für alle Elemente innerhalb der MIME-Mail-Struktur, die ein „body“-Element darstellen, ebenfalls statt des Feldes „data“

Der Client darf an folgenden Stellen „tempid“-Felder senden:

- PIM-Attachments: Innerhalb eines Objekts der im Feld „attachments“ gesendeten Liste von Attachment-Objekten, statt des Feldes „data“
- Emails: Für alle Elemente innerhalb der MIME-Mail-Struktur, die ein „body“-Element darstellen, ebenfalls statt des Feldes „data“
- S/MIME-Emails: Innerhalb des Mail-Objekts, anstelle des Feldes „smime_body_data“

Einschränkungen bei der Nutzung von „tempid“-Daten

Die Nutzung von „tempid“ und den dazugehörigen Befehlen (s.u.) unterliegt gewissen technischen und sicherheitsrelevanten Einschränkungen:

- Eine „tempid“ muss einen gültigen UTF-8-String darstellen
- Vom Server generierte „tempid“ haben üblicherweise eine Länge von unter 100 Zeichen (kann bei sehr langen Email-Folder-Namen oder –Hierarchien überschritten werden)
- Vom Server gemeldete „tempid“ sind so lange gültig, wie sich das zugehörige Objekt auf dem Server nicht geändert hat (bei einem entsprechenden Fehler von

downloadData muss ein neuer syncUpdate angestossen werden, um die Server-Änderungen und ggfs. neue „tempid“ zu erhalten):

- PIM-Attachments: Attachment wurde nicht geändert/gelöscht und enthaltendes Objekt befindet sich noch im gleichen Folder
- Email-Attachments: Mail wurde nicht geändert, verschoben oder gelöscht
- Vom Client generierte „tempid“ dürfen eine Länge von 16k Bytes nicht überschreiten (Sicherheits-Limit). Sie sollten so kurz wie möglich sein.
- Client-generierte „tempid“ werden spezifisch für diesen Client gespeichert, d.h. es muss nicht auf globale Eindeutigkeit geachtet werden (z.B. ist eine einfache Sequenz „1“, „2“, „3“ möglich)
- Alle vom Client gesendeten temporären Daten werden nach dem nächsten synclnit-, syncUpdate-, moveItems-, sendMail- oder logout-Befehl auf dem Server gelöscht
- Alle vom Client gesendeten temporären Daten werden nach einem auf dem Server konfigurierten Intervall der Inaktivität (individuell pro „tempid“, default: 5 Minuten) automatisch gelöscht. Dieses Intervall wird bei jedem Zugriff über uploadData für den dabei referenzierten Datensatz zurückgesetzt. Falls der Client mehrere sehr große Datensätze für einen Vorgang zum Server übertragen muss, kann er sicherstellen, dass alle bereits hochgeladenen Daten lange genug auf dem Server vorgehalten werden, indem er regelmässig einen uploadData-Request mit leerem Dateninhalt durchführt.
- Wenn eine vom Client referenzierte „tempid“ auf dem Server bei der Ausführung eines Befehls nicht aufgelöst werden kann (Daten nicht oder nicht mehr vorhanden), bricht die Aktion (sendMail) bzw. die Teil-Operation für dieses Object (Attachment- oder Mail-Erzeugung bei synclnit/syncUpdate) mit einem Fehler ab.

Download Data

Pfad:

<Basis-URL>/downloadData

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
tempid	String	Nein		Vom Server in einem früheren Befehl gemeldete „tempid“
offset	Integer	Ja	0	Offset (in Bytes), ab dem die Daten gelesen werden sollen
length	Integer	Ja	(unbegrenzt)	Maximale Anzahl in Bytes der zu lesenden Daten

Response:

Feld	Datentyp	Optional	Beschreibung
data	String	Nein	Base64-kodierter (Teil des) Datensatz, der von „tempid“ referenziert wird
moreAvailable	Boolean	Ja	OX-ID des Contexts des OX-Servers, in dem der Benutzer registriert ist

Beschreibung:

Mit diesem Befehl kann der Client Daten vom Server laden, die von diesem über eine „tempid“ referenziert wurden. Es ist nur ein Download von durch den Server erzeugten „tempid“ möglich, vom Client erzeugte „tempid“ werden ignoriert und erzeugen einen Fehler (um Missbrauch des Servers als externe Storage auszuschliessen).

Falls ein Fehler bereits vor dem Beginn der Datenübertragung festgestellt wird (z.B. ungültige „tempid“ oder Objekt wurde zwischenzeitlich verändert), wird eine Standard-Fehlerantwort inkl. Fehlerdetails (soweit vorhanden) geliefert.

Tritt ein Fehler während der Übertragung der Daten auf, sind die ab diesem Zeitpunkt gesendeten Daten ungültig (es werden weiterhin Daten, genauer 0-Bytes gesendet, um die vom Client erwartete Content-Length zu erreichen) und es wird der Status-Code 2 (OX-Server-Error) gemeldet. Der Client muss in diesem Fall alle in diesem Request erhaltenen Daten als ungültig ansehen und verwerfen.

Bei einem Fehler muss der Client den Befehl, welcher die „tempid“ geliefert hat, wiederholen, um die aktuellen Server-Änderungen zu erhalten.

Beispiel:

Request:

```
PUT http://localhost/usm/downloadData
Content-Type: text/javascript

{"sessionId":"USM-
1","tempid":"server_temp_id","offset":50,"length":6}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"data":{"moreAvailable":true,"data":"AAAAAA=="}, "status":1}
```

Upload Data

Pfad:

<Basis-URL>/uploadData

Parameter:

Da der Server die vom Client gesendeten Daten direkt in einem temporären Bereich speichert, ohne das potentiell große JSON-Objekt komplett in den Speicher zu laden, ist für diesen Befehl eine definierte Reihenfolge aller Parameter notwendig. Um dies sicherzustellen, werden die Parameter nicht in einem JSON-Objekt, sondern in einem JSON-Array in folgender Reihenfolge erwartet:

Datentyp	Optional	Default Wert	Beschreibung
String	Nein		Session-ID
String	Nein		Vom Client definierte „tempid“
Integer	Ja	0	Offset (in Bytes), ab dem die Daten geschrieben werden sollen (64-bit Integer!)
String	Nein		Base64-kodierter (Teil des) Datensatz, der von „tempid“ referenziert werden soll und ab dem spezifizierten Offset geschrieben werden soll

Response:

Feld	Datentyp	Optional	Beschreibung
original_size	Integer	Nein	Größe des auf dem Server vorhandenen Datensatzes, bevor die vom Client hochgeladenen Daten hinzugefügt wurden (64-bit Integer!)

Beschreibung:

Mit diesem Befehl kann der Client Daten auf dem Server speichern, die dann später über die angegebene „tempid“ referenziert werden. Vom Client erzeugte „tempid“ werden getrennt von Server-„tempid“ verwaltet, es ist also unmöglich, dass über „uploadData“ vom Server gemeldete Daten überschrieben werden.

Im Erfolgsfall liefert der Server den Status 1 (Success) zurück, im Feld „original_size“ wird dir Größe des Datensatzes auf dem Server **vor** dem Hochladen der neuen Dateien geliefert. Falls keine Daten unter dieser „tempid“ hinterlegt waren, wird 0 geliefert. Mittels dieses Wertes kann der Client die Konsistenz der hochgeladenen Daten überprüfen. original_size muss jeweils gleich dem Maximum aus dem zuletzt gemeldeten original_size und dem zuletzt gesendeten Offset plus der Menge der übertragenen Daten sein:

$$\text{new_original_size} == \text{MAX}(\text{old_original_size}, \text{offset} + \text{length})$$

Der erwartete original_size-Wert entspricht bei einem sequentiellen Schreiben in mehreren Schritten dem gesendeten Offset. Der Client kann damit feststellen, ob z.B. die temporären Daten mittlerweile vom Server wegen eines Timeouts gelöscht wurden.

Im Fehlerfall wird eine Standard-Fehlerantwort zurückgegeben. Falls ein Fehler gemeldet wird, muss der Client davon ausgehen, dass die unter dieser „tempid“ hinterlegten Daten nicht mehr gültig oder korrupt sind. Vor einem erneuten Upload der Daten sollte entweder eine neue „tempid“ vergeben werden, oder die Daten explizit mittels „clearData“ gelöscht werden. Daten, die vom Client unter anderen „tempid“ gespeichert wurden, sind bei einem Fehler nicht betroffen.

Wenn ein Client ein uploadData mit einem Offset durchführt, der größer ist als die bereits unter der „tempid“ hinterlegten Daten, wird der Datensatz entsprechend vergrößert, so dass ab dem angegebenen Offset geschrieben werden kann. Der Inhalt der Daten zwischen bisheriger Größe und angegebenen Offset ist als undefiniert anzusehen, kann aber vom Client zu einem späteren Zeitpunkt überschrieben werden.

Vom Client hochgeladene Daten werden vom Server nur für einen begrenzten Zeitraum gespeichert, solange keine weiteren Daten hinzugefügt werden (default: 5 Minuten). Wenn mehrere große Datensätze hochgeladen werden sollen, kann der Client bereits hochgeladene Datensätze als noch in Benutzung markieren, indem er für die „tempid“ regelmässig ein uploadData mit einem leeren Datensatz durchführt (d.h. [“sessionid“,“tempid“,““])

Beispiel:

Request:

```
PUT http://localhost/usm/uploadData
Content-Type: text/javascript

["USM-1","client_temp_id",50,"AAAAAA=="]
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1,"data":{"original_size":50}}
```

Clear Data

Pfad:

<Basis-URL>/clearData

Parameter:

Name	Datentyp	Optional	Default Wert	Beschreibung
sessionid	String	Nein		
tempid	String	Ja		Vom Client in einem früheren Befehl gesendete „tempid“

Response:

leer

Beschreibung:

Mit diesem Befehl kann der Client temporäre Daten auf dem Server löschen, die er vorher über eine „tempid“ erzeugt hat. Wird der optionale Parameter „tempid“ angegeben, werden nur die temporären Daten zu dieser „tempid“ entfernt. Ohne Angabe des Parameters „tempid“ werden alle vom Client erzeugten temporären Daten gelöscht.

Falls ein interner Fehler beim Zugriff auf die Daten auftritt, wird eine Standard-Fehlerantwort geliefert. Wenn die Datei(en) gelöscht wurde(n) oder wenn unter der angegebenen „tempid“ keine temporären Daten auf dem Server (mehr) vorhanden sind, wird der Status 1 (Success) gemeldet.

Nach der Ausführung eines syncInit, syncUpdate, moveItems-, sendMail- oder logout-Befehls (unabhängig ob erfolgreich oder nicht) werden vom Server automatisch alle vom Client angelegten temporären Daten gelöscht. Es ist also im Normalfall nicht notwendig, vom Client erzeugte temporäre Daten explizit zu löschen.

Beispiel:

Request:

```
PUT http://localhost/usm/clearData
Content-Type: text/javascript

{"sessionId":"USM-1","tempid":"client_temp_id"}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/javascript

{"status":1}
```

Erweiterungen USM-JSON-API (Version 2.1.0)

In Version 2.1.0 der USM-JSON-API werden die Erweiterungen aus der Version 2.0.0 fortgeführt. Zusätzlich zum Streaming von PIM- und Mail-Attachments wird nun auch ein Streaming von Kontakt-Bildern implementiert. Es werden die gleichen Mechanismen wie für Attachments verwendet:

- Mittels „UploadData“ kann ein Bild vor einer Synchronisation unter einer temporären ID „tempid“ vom Client hochgeladen werden.
- Statt des Feldes „image1“ **kann** vom Client das Feld „tempid“ gesendet werden, welches vorher temporär hochgeladene Daten referenziert (die direkte Verwendung von „image1“ durch den Client ist immer noch erlaubt).
- Der Server sendet statt „image1“ in Version 2.1.0 **immer** das Feld „tempid“. Dies wird durchgeführt, da die Bildgröße nicht vor dem Download ermittelbar ist, und da der Server unter Umständen auch eine Konvertierung des Bildes ins JPEG-Format durchführen muss.
- Mittels „DownloadData“ kann der Client ein Kontakt-Bild unter der vom Server referenzierten „tempid“ herunterladen. Dabei wird das Bild automatisch in das JPEG-Format konvertiert.
- Die in Version 2.0.0 eingeführten Befehle „UploadData“, „DownloadData“, „ClearData“ werden für die Daten von Kontakt-Bildern genauso genutzt wie für Attachments. Auch das implizite Server-Verhalten (Löschung nach konfiguriertem Zeitintervall, Löschung nach Synchronisations-Befehlen) ist gleich.

OX-JSON-CALLS

Pfad:

<Basis-URL>/oxcall

Der Request-Body ist immer ein JSON Objekt, in dem immer mindestens die folgenden Feldern gesetzt sein müssen:

- **module** (String) benennt das entsprechende OX-Modul (z.B. login, contacts, tasks, mail, folders)

- **action** (String) benennt den auszuführenden OX-Befehl

Weitere notwendige Felder sind bei den einzelnen Befehlen beschrieben. Als Response wird ein wie in der Einleitung beschriebenes JSON Objekt zurückgeliefert, die Antwort-Daten werden im Feld „data“ übergeben.

Modul: folders

Get root folders

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	folders	
action	String	Nein	root	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-Nummern, die abgefragt werden sollen

Response:

JSONArray. Enthält alle Folders auf Root-Ebene.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_root_folders

Get subfolders

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	folders	
action	String	Nein	list	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
parent	String	Nein		Parent folder id

Response:

JSONArray mit allen Folders, die als parent den übergebenen Folder haben

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_subfolders

Get path**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	folders	
action	String	Nein	path	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-Nummern, die abgefragt werden sollen
id	String	Nein		Die Objekt ID des Folders

Response:

JSONArray mit allen Parent-Folders aufwärts bis zum root-Folder.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_path

Get updated folders**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	folders	
action	String	Nein	updates	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-Nummern, die abgefragt werden sollen
parent	String	Nein		Parent folder id
timestamp	Long	Nein		letztes Update
ignore	String	Ja	deleted	Falls vorhanden, gelöschte Objekte werden

				nicht zurückgeliefert
--	--	--	--	-----------------------

Response:

JSONArray mit Daten für neue, modifizierte und gelöschte Folders. Die neue und modifizierte Folders werden in Listen (arrays) zurückgegeben. Die gelöschten Folders werden als einfache Strings (die IDs) zurückgegeben und sind nicht Teil einer verschachtelten Liste.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_updated_folders

Get a folder

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	folders	
action	String	Nein	get	
session	String	Nein		
id	String	Nein		Die Objekt ID des Folders

Response:

JSONObject mit dem Inhalt des Folders außer dem Feld id

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_folder

Create a folder

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	folders	
action	String	Nein	new	
session	String	Nein		
folder_id	String	Nein		Parent folder id
dataobject	JSONObject	Nein		das Folder-Objekt als JSONObject (ohne das Feld id)

Response:

JSONObject mit dem Attribut objectid für das neu erzeugte Folder-Objekt

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Create_a_folder

Update a folder**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	folders	
action	String	Nein	update	
session	String	Nein		
timestamp	Long	Nein		Letztes Update
id	String	Nein		Die Objekt ID des Folders
dataobject	JSONObject	Nein		das Folder-Objekt als JSONObject (ohne das Feld id)

Response:

JSONObject, timestamp (Long)

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Update_a_folder

Delete folders**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	folders	
action	String	Nein	delete	
session	String	Nein		
timestamp	Long	Nein		Letztes Update

objectids	JSONArray	Nein		Liste der Ids von Folders, die gelöscht werden sollen
-----------	-----------	------	--	---

Response:

JSONObject mit JSONArray, in dem ids von Folders, die nach dem mitgegebenen Timestamp geändert wurden und deswegen nicht gelöscht werden konnten, stehen.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_folders

Modul: tasks

Get all tasks

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	all	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-Nummern, die abgefragt werden sollen
folder	String	Nein		ID des Task-Folders
sort	Integer	Ja		Nummer des Felds, das die Sortierung des Ergebnisses bestimmen soll. Wenn der Parameter sort angegeben ist, muss auch der Parameter order angegeben werden.
order	String	Ja	asc / desc	„asc“ – aufsteigend; „desc“ - absteigend

Response:

JSONObject mit JSONArray, welches die angeforderten Task-Objekte enthält. Jedes Array-Element beschreibt ein Task-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_all_tasks

Get a list of tasks

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	list	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-Nummern, die abgefragt werden sollen
objectids	JSONArray	Nein		Liste von angeforderten Task-Ids

Response:

JSONObject mit JSONArray, in dem die angeforderten Task-Objekte enthalten sind. Jedes Array-Element beschreibt ein Task-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_list_of_tasks

Get updated tasks

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	updates	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-Nummern, die abgefragt werden sollen
folder	String	Nein		ID des Task-Folders
sort	Integer	Ja		ID des Felds, das die Sortierung des Ergebnisses bestimmen soll. Wenn der Parameter sort gesetzt ist, muss auch der Parameter order angegeben werden.
order	String	Ja	asc / desc	„asc“ – aufsteigend; „desc“ - absteigend
timestamp	Long	Nein		letztes Update

Response:

JSONArray mit Daten für neue, modifizierte und gelöschte Tasks. Die neuen und modifizierten Tasks werden als Listen (arrays) zurückgegeben. Die gelöschten Tasks werden als einfache Strings (die IDs) zurückgegeben und sind nicht Teil einer verschachtelten Liste.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_updated_tasks

Get a task**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	get	
session	String	Nein		
folder	String	Nein		ID des Task-Folders
id	String	Nein		ID des angeforderten Task-Objektes

Response:

JSONObject mit allen Daten aus dem angeforderten Task-Objekt außer dem id Feld

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_task

Create a task**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	new	
session	String	Nein		
dataobject	JSONObject	Nein		Task als JSONObject

Response:

JSONObject mit dem Attribut objectid für neu erzeugtes Task-Objekt

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Create_a_task

Update a task

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	update	
session	String	Nein		
folder	String	Nein		ID des enthaltenden Task-Folders
id	String	Nein		ID des zu ändernden Task-Objektes
dataobject	JSONObject	Nein		Task als JSONObject
timestamp	Long	Nein		Timestamp der letzten Änderung. Wenn der Task nach dem spezifizierten Timestamp geändert wurde, wird das Update abgebrochen

Response:

JSONObject, timestamp (Long)

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Update_a_task

Delete tasks

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	delete	
session	String	Nein		
timestamp	Long	Nein		letztes Update
objectids	JSONArray	Nein		Liste von JSONObject, jedes JSONObject muss das Feld id (Task-Id) und das Feld parentfolder (mit id für Parent-Folder) enthalten

Response:

JSONObject mit JSONArray, in dem die IDs von Tasks, gelistet sind, die nach dem mitgegebenen Timestamp geändert wurden und deswegen nicht gelöscht werden konnten.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_tasks

Confirm a task

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	confirm	
session	String	Nein		
timestamp	Long	Nein		letztes Update
id	String	Nein		Id des Task-Objektes
folder	String	Nein		Id des Folders, wo der Task sich befindet
requestBody	JSONObject	Nein		JSON Objekt mit 2 Felder: confirmation und confirmmessage (wie in der Tabelle User participant object im Anhang B)

Response:

JSONObject, timestamp (Long)

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Confirm_task

Search for tasks

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	tasks	
action	String	Nein	search	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
timestamp	Long	Nein		letztes Update

sort	Integer	Ja		Id des Felds, das die Sortierung des Ergebnisses bestimmen soll. Wenn der Parameter sort gegeben ist, muss auch der Parameter order gegeben werden.
order	String	Ja	asc / desc	„asc“ – aufsteigend; „desc“ - absteigend
requestBody	JSONObject	Nein		die Felder sind in der Tabelle „Search Tasks Object Fields“ beschrieben

Response:

JSONObject mit JSONArray, wo angeforderten Task – Objekte gespeichert sind. Jedes Array –Element beschreibt ein Task-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Search_for_tasks

Search Tasks Object Fields

Name	Type	Value
pattern	String	Search pattern to find tasks. In the pattern, the character "*" matches zero or more characters and the character "?" matches exactly one character. All other characters match only themselves.
folder	Number	(optional) Defines the folder to search for tasks in. If this is omitted in all task folders will be searched.
start	Date or Time	(optional) Inclusive start date for a time range the tasks should end in. If start is omitted end is ignored.
end	Date or Time	(optional) Exclusive end date for a time range the tasks should end in. If this parameter is omitted the time range has an open end.

Modul: contacts

Get all contacts

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
------	----------	----------	------	-----------

module	String	Nein	contacts	
action	String	Nein	all	
session	String	Nein		
folder	String	Nein		Id des Folders, wo das Contact sich befindet
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
sort	Integer	Ja		Id des Felds, das die Sortierung des Ergebnisses bestimmen soll. Wenn der Parameter sort gegeben ist, muss auch der Parameter order gegeben werden.
order	String	Ja	asc / desc	„asc“ – aufsteigend; „desc“ - absteigend

Response:

JSONObject mit JSONArray, wo angeforderten Contact – Objekte gespeichert sind. Jedes Array –Element beschreibt ein Contact-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_all_contacts

Get a list of contacts

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	
action	String	Nein	list	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
objectids	JSONArray	Nein		Liste von angeforderten Contact-Ids

Response:

JSONObject mit JSONArray, wo angeforderten Contact – Objekte gespeichert sind. Jedes Array – Element beschreibt ein Contact-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_list_of_contacts

Get a list of users

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	
action	String	Nein	listuser	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
objectids	JSONArray	Nein		Liste von angeforderten Contact-Ids

Response:

JSONObject mit JSONArray, wo angeforderten Contact – Objekte gespeichert sind. Jedes Array –Element beschreibt ein Contact-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_list_of_users

Get updated contacts

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	
action	String	Nein	updates	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
folder	String	Nein		Id des Folders, wo das Contacts sich befindet
sort	Integer	Ja		Id des Felds, das die Sortierung des Ergebnisses bestimmen soll. Wenn der Parameter sort gegeben ist, muss auch der Parameter order gegeben werden.
order	String	Ja	asc / desc	„asc“ – aufsteigend; „desc“ - absteigend

timestamp	Long	Nein		letztes Update
-----------	------	------	--	----------------

Response:

JSONArray mit Daten für neue, modifizierte und gelöschte Contacts. Die neuen und modifizierten Contacts werden als Listen (arrays) zurückgegeben. Die gelöschten Contacts werden als einfache Strings (die IDs) zurückgegeben und sind nicht Teil einer verschachtelten Liste.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_updated_contacts

Get a contact

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	
action	String	Nein	get	
session	String	Nein		
id	String	Nein		Id des angeforderten Contact-Objektes
folder	String	Nein		Id des Folders, in dem sich das Contact befindet

Response:

JSONObject mit allen Daten aus dem angeforderten Contact-Objekt außer dem id –Feld

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_contact

Get a contact by user ID

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	
action	String	Nein	getuser	

session	String	Nein		
id	String	Nein		User ID des angeforderten Contact-Objektes (nicht Objekt- ID!)

Response:

JSONObject mit allen Daten aus dem angeforderten Contact-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_contact_by_user_ID

Create a contact

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	
action	String	Nein	new	
session	String	Nein		
dataobject	JSONObject	Nein		DataObject vom ContactContentType als JSONObject

Response:

JSONObject mit dem Attribut id, wo eine Id für das neu erzeugtes Contact-Objekt steht.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Create_a_contact

Update a contact

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	
action	String	Nein	update	

session	String	Nein		
id	String	Nein		Id des angeforderten Contact-Objektes
folder	String	Nein		Id des Folders, in dem sich das Contact-Objekt befindet
timestamp	Long	Nein		Timestamp vom modifizierten Contact. Wenn das Contact nach dem spezifizierten Timestamp geändert wurde, wird das Update abgebrochen.

Response:

JSONObject, timestamp (Long)

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Update_a_contact

Delete contacts

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	
action	String	Nein	delete	
session	String	Nein		
timestamp	Long	Nein		letztes Update
objectids	JSONArray	Nein		JSONArray von JSONObject, wo jedes JSONObject das Feld id (Contact-Id) und das Feld parentfolder (mit id für Parent-Folder) enthält

Response:

JSONObject mit JSONArray, wo ids von Contacts, die nach dem mitgegebenen Timestamp geändert wurden und deswegen nicht gelöscht werden konnten, stehen.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_contacts

Search contacts

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	contacts	

action	String	Nein	search	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
sort	Integer	Ja		Id des Felds, das die Sortierung des Ergebnisses bestimmen soll. Wenn der Parameter sort gegeben ist, muss auch der Parameter order gegeben werden.
order	String	Ja	asc / desc	„asc“ – aufsteigend; „desc“ – absteigend
requestBody	JSONObject	Nein		die Felder sind in der Tabelle „Search contacts Object Fields“ oder „Search contacts Object Fields alternative“ beschrieben

Response:

JSONObject mit JSONArray, wo angeforderten Contact – Objekte gespeichert sind. Jedes Array – Element beschreibt ein Contact-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Search_contacts

Search Contacts Object Fields

Name	Type	Value
pattern	String	Search pattern to find contacts. In the pattern, the character "*" matches zero or more characters and the character "?" matches exactly one character. All other characters match only themselves.
startletter	String	Search contacts with the given startletter. If this field is present, the pattern is matched against the contact field which is specified by the property contact_first_letter_field on the server (default: last name). Otherwise, the pattern is matched against the display name.
folder	Array of Number	If a list of folder identifiers or at least a single folder identifier is given, only in that folders will be searched for contacts. This parameter is optional but searching in all contact folders that are viewable and where objects can be read in is more expensive on that database than searching in a dedicated number of them.

Search Contacts Object Fields alternative

Name	Type	Value
last_name	String	Searches contacts where the last name match with the given last name.

first_name	String	Searches contacts where the first name match with the given first name.
display_name	String	Searches contacts where the display name match with the given display name.
orSearch	Boolean	If set to true, the fields are connected through an OR search habit.
emailAutoComplete	Boolean	If set to true, results are guaranteed to contain at least one email adress and the search is performed by connecting the relevant fields through an OR search habit.

Modul: calendar

Get all appointments

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	all	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
folder	String	Ja		Id des Folders, in dem Appointments zu finden sind. Wenn nicht gegeben, alle Calendar Folders werden abgefragt.
start	Date	Nein		Nur Termine, die an dem gegebenen Start Datum oder später stattfinden, werden zurückgeliefert
end	Date	Nein		Nur Termine, die bis dem gegebenen End Datum (ausschließlich) enden, werden zurückgeliefert
recurrence_master	Boolean	Nein		Wenn auf true gesetzt, dann werden alle einzelne Termine einer Serie als einzelne Objekte zurückgeliefert

Response:

JSONObject mit JSONArray, wo angeforderten Calendar – Objekte gespeichert sind. Jedes Array –Element beschreibt ein Calendar-Objekt (Termin). Die Termine sind in aufsteigende Reihenfolge nach dem Feld start_date sortiert.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_all_appointments

Get a list of appointments

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	list	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
recurrence_master	Boolean	Nein		Wenn auf true gesetzt, dann werden alle einzelne Termine einer Serie als einzelne Objekte zurückgeliefert

Response:

JSONObject mit JSONArray, wo angeforderten Calendar – Objekte gespeichert sind. Jedes Array – Element beschreibt ein Calendar-Objekt (Termin).

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_list_of_appointments

Get updated appointments

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	updates	
session	String	Nein		
folder	String	Nein		Id des Folders, in dem Appointments zu finden sind.
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
start	Date	Nein		Nur Termine, die an dem gegebenen Start Datum oder später stattfinden, werden zurückgeliefert
end	Date	Nein		Nur Termine, die bis dem gegebenen End Datum (ausschließlich) enden, werden zurückgeliefert

recurrence_master	Boolean	Nein		Wenn auf true gesetzt, dann werden alle einzelne Termine einer Serie als einzelne Objekte zurückgeliefert
-------------------	---------	------	--	---

Response:

JSONArray mit Daten für neue, modifizierte und gelöschte Appointments. Die neuen und modifizierten Appointments werden als Listen (arrays) zurückgegeben. Die gelöschten Appointments werden als einfache Strings (die IDs) zurückgegeben und sind nicht Teil einer verschachtelten Liste.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_updated_appointments

Get appointment information

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	has	
session	String	Nein		
start	Date	Nein		Nur Termine die an dem gegebenen Start Datum oder später stattfinden werden zurückgeliefert
end	Date	Nein		Nur Termine die bis dem gegebenen End Datum (ausschließlich) enden werden zurückgeliefert

Response:

JSONArray mit Boolean Werten. Jeder Wert entspricht einem Tag, der abgefragt worden ist, und zeigt, ob es an dem bestimmten Tag einen Termin gibt oder nicht.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_appointment_information

Get an appointment

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	get	

session	String	Nein		
id	String	Nein		Id des Appointment-Objektes
folder	Boolean	Nein		Id des Folders, in dem der Termin sich befindet.
recurrence_position	Integer	Ja		Die Position des Termins in einer Serie. Benötigt nur wenn man einen einzelnen Termin von einer Serie abfragen will.

Response:

JSONObject mit allen Daten aus dem angeforderten Calendar-Objekt außer dem id Feld

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_an_appointment

Create an appointment

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	new	
session	String	Nein		
dataobject	JSONObject	Nein		das Appointment-Objekt als JSONObject, ohne das Feld id.

Response:

If the appointment was created successfully, an object with the attribute id of the newly created appointment. If the appointment could not be created due to conflicts, the response body is an object with the field conflicts, which is an array of appointment objects which caused the conflict. Each appointment object which represents a resource conflict contains an additional field hard_conflict with the Boolean value true. If the user does not have read access to a conflicting appointment, only the fields id, start_date, end_date, shown_as and participants are present and the field participants contains only the participants which caused the conflict.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Create_an_appointment

Update an appointment

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	

action	String	Nein	update	
session	String	Nein		
folder	String	Nein		Id des Folders, in dem der Termin sich befindet.
timestamp	String	Nein		Timestamp vom modifizierten Termin. Wenn der Termin nach dem spezifizierten Timestamp geändert wurde, wird das Update abgebrochen.
dataobject	JSONObject	Nein		das Appointment-Objekt als JSONObject ohne das Feld id. Das Feld recurrence_id ist immer gesetzt. Das Feld recurrence_position ist gesetzt nur wenn man einen einzelnen Termin von einer Serie ändern will.

Response:

JSONObject, timestamp (Long)

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Update_an_appointment

Delete appointments

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	delete	
session	String	Nein		
timestamp	String	Nein		Timestamp vom modifizierten Termin. Wenn der Termin nach dem spezifizierten Timestamp geändert wurde, wird das Update abgebrochen.
dataobject	JSONObject	Nein		das Appointment-Objekt als JSONObject

Response:

JSONObject mit JSONArray, wo ids von Appointments, die nach dem mitgegebenen Timestamp geändert wurden und deswegen nicht gelöscht werden konnten, stehen.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_appointments

Confirm appointment

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	confirm	
session	String	Nein		
id	String	Nein		Object ID of the appointment to confirm.
folder	String	Nein		ID of the folder through which the appointment is accessed.
timestamp	String	Nein		Timestamp of the last update of the to confirmed appointment.
requestBody	JSONObject	Nein		(JSONObject) JSON Objekt mit 2 Felder: confirmation und confirmmessage (wie in der Tabelle „User participant object“ im Anhang B)

Response:

JSONObject, timestamp (Long)

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Confirm_appointment

Search appointments

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	search	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen

requestBody	JSONObject	Nein		die Felder sind in der Tabelle „Search appointments Object Fields“ beschrieben
-------------	------------	------	--	--

Response:

JSONObject mit JSONArray, wo angeforderten Appointment – Objekte gespeichert sind.
Jedes Array –Element beschreibt ein Appointment-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Search_appointments

Search Appointments Object Fields

Name	Type	Value
pattern	String	Search pattern to find appointments. In the pattern, the character "*" matches zero or more characters and the character "?" matches exactly one character. All other characters match only themselves.
startletter	String	Search appointments with the given starting letter.

Get new appointments

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	newappointments	
session	String	Nein		
columns	String	Nein		Liste von durch Komma getrennten Feld-IDs, die abgefragt werden müssen
start	Date	Nein		Nur Termine, die an dem gegebenen Start Datum oder später stattfinden, werden zurückgeliefert
end	Date	Nein		Nur Termine, die bis dem gegebenen End Datum (ausschließlich) enden, werden zurückgeliefert
sort	Integer	Ja		The identifier of a column which determines the sort order of the response. If this parameter is specified and holds a column number, then the parameter order must be also specified.

order	String	Ja		"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.
limit	Integer	Nein		limits the number of returned object to the given value.

Response:

JSONObject mit JSONArray, wo angeforderten Appointment – Objekte gespeichert sind.
Jedes Array – Element beschreibt ein Appointment-Objekt.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_new_appointments

Free & Busy

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	calendar	
action	String	Nein	freebusy	
session	String	Nein		
id	String	Nein		Internal user id. Must be obtained from the contact module.
type	Date	Nein		Constant for user or resource
start	Date	Nein		Lower inclusive limit of the queried range as a Date. Only appointments which end on or after this date are returned.
end	Date	Nein		Upper exclusive limit of the queried range as a Date. Only appointments which start before this date are returned.

Response:

An array of objects identifying the appointments which lie between start and end as described. This objects consist of:

Name	Type	Value
shown_as	Number	Describes, how this appointment appears in availability queries: 1 reserved

2 temporary

3 absent

4 free

start_date	Date or Time	see Detailed task and appointment data
end_date	Date or Time	see Detailed task and appointment data
id	String	Object ID
folder_id	String	Folder ID. Only set, if the user has the right to see the object. (added 2009-08-18/6.12)
full_time	Boolean	True if the appointment is a whole day appointment, not present otherwise.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Free_.26_Busy

Modul: mail

Get mail count

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	count	
session	String	Nein		
folder	String	Nein		Object ID of the folder whose mail count is queried

Response:

An integer value representing folder's mail count

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_mail_count

Get all mails

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	

action	String	Nein	all	
session	String	Nein		
folder	String	Nein		Object ID of the folder whose mail count is queried
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for appointments are defined in the table Detailed mail data in „Anhang B“
sort	Integer	Ja		The identifier of a column which determines the sort order of the response or the string "thread" to return thread-sorted messages. If this parameter is specified and holds a column number, then the parameter order must be also specified.
order	String	Ja	asc / desc	"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.

Response:

An array with mail data. Each array element describes one mail and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_all_mails

Search mails

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	search	
session	String	Nein		
folder	String	Nein		Object ID of the folder whose mail count is queried
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for appointments are defined in the table Detailed mail data in „Anhang B“
sort	Integer	Ja		The identifier of a column which determines the sort order of the response or the string "thread" to return thread-sorted messages. If

				this parameter is specified and holds a column number, then the parameter order must be also specified.
order	String	Ja	asc / desc	"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.
requestBody	JSONArray	Nein		A JSON array of JSON objects each containing the search field and its search pattern: e.g.: [{"col": 612, "pattern": "Joe"}, {"col": 614, "pattern": "Tuesday"}]

Response:

An array with mail data. Each array element describes one mail and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Search_mails

Get a list of mails

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	list	
session	String	Nein		
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for appointments are defined in the table Detailed mail data in „Anhang B“
objectids	JSONArray	Nein		An array with one object for each requested mail. Each object contains the fields folder and id.

Response:

An array with mail data. Each array element describes one mail and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_list_of_mails

Copy mails

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	copy	
session	String	Nein		
id	String	Nein		Object ID of the requested mail.
folder	String	Nein		Object ID of the source folder.
requestBody	JSONObject	Nein		A JSON object containing the id of the destination folder inside the "folder_id" field: e.g.: {"folder_id": 1376}

Response:

A JSON array containing the ID of the copied mail

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Copy_mails

Move mails

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	update	
session	String	Nein		
id	String	Nein		Object ID of the requested mail.
folder	String	Nein		Object ID of the source folder.
requestBody	JSONObject	Nein		A JSON object containing the id of the destination folder inside the "folder_id" field: e.g.: {"folder_id": 1376}

Response:

A JSON array containing the ID of the copied mail

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Move_mails

Update mails

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	update	
session	String	Nein		
id	String	Nein		Object ID of the requested mail.
folder	String	Nein		Object ID of the source folder.
requestBody	JSONObject	Nein		A JSON object which carries the new values that ought to be applied to mail as described in Update mail or Update mail extended tables

Response:

A JSON object containing the Object ID of the updated mail and its folder.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Update_mails

Update Mail

Name	Type	Value
color_label	Number	The color number between 1 and 10.
flags	Number	A set of flags to add or remove. Note: Flags for "recent" (8) and "user" (64) are ignored.
value	Boolean	true to add the flags specified by flags (logical OR), false to remove them (logical AND with the inverted value).

Update mail extended (available with SP6 v6.10)

Name	Type	Value
set_flags	Number	A set of flags to add. Note: Flags for "recent" (8) and "user" (64) are ignored.
clear_flags	Number	A set of flags to remove. Note: Flags for "recent" (8) and "user" (64) are ignored.

Get a mail

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	get	
session	String	Nein		
id	String	Nein		Object ID of the requested mail.
folder	String	Nein		Object ID of the mail's folder.
edit	String	Ja	1	indicates that this request should fill the message compose dialog to edit a message and thus display-specific date is going to be withheld.
hdr	String	Ja	1	to let the response contain only the (formatted) message headers as plain text
src	String	Ja	1	to let the response contain the complete message source as plain text
save	String	Ja	1	to write the complete message source to output stream. NOTE: This parameter will only be used if parameter src is set to 1.
view	String	Ja		"raw" returns the content as it is, meaning no preparation are performed and thus no guarantee for safe contents is given (available with SP6 v6.10). "text" forces the server to deliver a text-only version of the requested mail's body, even if content is HTML. "html" to allow a possible HTML mail body being transferred as it is (but white-list filter applied). "noimg" to allow a possible HTML content being transferred but without original image src attributes which references external images: Can be used to prevent loading external linked images (spam privacy protection). NOTE: if set, the corresponding gui config setting will be ignored.
unseen	String	Ja		"1" or "true" to leave an unseen mail as unseen although its content is requested

Response:

An JSON object containing all data of the requested mail. The fields of the object are listed in Detailed mail data. The fields id and attachment are not included. **NOTE:** Of course response is not a JSON object if either parameter hdr or parameter src are set to "1". Then the response contains plain text.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_mail

Get a mail attachment

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	attachment	
session	String	Nein		
id	String	Nein		Object ID of the mail which contains the attachment.
attachment	String	Ja		ID of the requested attachment OR
cid	String	Ja		Value of header 'Content-ID' of the requested attachment
save	String	Nein	1	overwrites the defined mimetype for this attachment to force the download dialog, otherwise 0.
filter	String	Ja		1 to apply HTML white-list filter rules if and only if requested attachment is of MIME type text/htm* AND parameter save is set to 0.

Response:

The raw byte data of the document. The response type for the HTTP Request is set accordingly to the defined mimetype for this attachment, except the parameter save is set to 1.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_mail_attachment

Send / Save mail**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	new	
session	String	Nein		
src	String	Nein	1	Set to "1" to accept mail message as MIME data block.

folder	String	Ja		In case the mail should not be sent out, but saved in a specific folder, the "folder" paramter can be used. If the mail should be sent out to the recipient, the "folder" parameter must not be included and the mail is stored in the folder "Sent Items". Example "folder=default.INBOX/Testfolder"
flags	Integer	Ja		In case the mail should be stored with status "read" (e.g. mail has been read already in the client inbox), the parameter "flags" has to be included. If no "folder" parameter is specified, this parameter must not be included. For infos about mail flags see Detailed mail data spec.
requestBody	String	Nein		The MIME Data Block

Response:

Object ID of the newly created/moved mail.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Send.2FSave_mail_as_MIME_data_block_.28RFC822.29_.28added_in_SP5.29

Reply / Forward a mail

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	reply replyall forward	
session	String	Nein		
id	String	Nein		Object ID of the requested Message.
folder	String	Nein		Object ID of the folder, whose contents are queried.
view	String	Ja		"text" forces the server to deliver a text-only version of the requested mail's body, even if content is HTML. "html" to allow a possible HTML mail body being transferred as it is (but white-list filter applied). NOTE: if set, the corresponding gui config

				setting will be ignored.
--	--	--	--	--------------------------

Response:

An object containing all data of the requested mail. The fields id and attachment are not included.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Reply.2FForward_a_mail

Delete mails

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	delete	
session	String	Nein		
timestamp	Long	Nein		Timestamp of the last update of the deleted mails.
objectids	JSONArray	Nein		An array with object IDs of the deleted mails.

Response:

An array with object IDs of mails which were modified after the specified timestamp and were therefore not deleted.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_mails

Clear mail folders

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	mail	
action	String	Nein	clear	
session	String	Nein		
timestamp	Long	Nein		Timestamp of the last update of the deleted

				mails.
objectids	JSONArray	Nein		An array with IDs of the mail folders to clear

Response:

An array with IDs of mail folder that could not be cleared; meaning the response body is an empty JSON array if everything went well.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Clear_mail_folder.28s.29

Module: groups

Get a group

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	groups	
action	String	Nein	get	
session	String	Nein		
id	String	Nein		The group id.

Response:

A group object as described in Group data in „Anhang B“.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_group

List groups

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	groups	
action	String	Nein	list	

session	String	Nein		
objectids	JSONArray	Nein		An array with group identifiers.

Response:

An array of group objects as described in Group data in „Anhang B“.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#List_groups

Search for groups

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	groups	
action	String	Nein	search	
session	String	Nein		
requestBody	JSONObject	Nein		An object with search parameters as described in the table "Group search object fields"

Response:

An array of group objects as described in Group data in „Anhang B“.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Search_for_groups

Group search object fields

Name	Type	Value
pattern	String	Search pattern to find groups. In the pattern, the character "*" matches zero or more characters and the character "?" matches exactly one character. All other characters match only themselves.

Create a group

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	groups	
action	String	Nein	new	
session	String	Nein		
dataobject	JSONObject	Nein		Group object as described in the table Group data in „Anhang B“. The field id is not present.

Response:

A json object with attribute id of the newly created group.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Create_a_group

Delete a group

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	groups	
action	String	Nein	delete	
session	String	Nein		
timestamp	Long	Nein		Timestamp of the last update of the group to delete.

Response:

A JSONObject with attribute id of the newly created group.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_a_group

Change a group

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	groups	
action	String	Nein	update	

session	String	Nein		
id	String	Nein		Object ID of the group to update.
timestamp	Long	Nein		Time stamp of the group to update. If the group was modified after the specified time stamp, then the update must fail.
dataobject	JSONObject	Nein		Group object as described in Group data. Only modified fields are present and the field id is omitted.

Response:

Timestamp of the updated group.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Change_a_group

Modul: resource

List resources

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	resource	
action	String	Nein	list	
session	String	Nein		
objectids	JSONArray	Nein		An array with resources ids.

Response:

An array of resource objects as described in the table Resource response below.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#List_resources

Resource response

Name	Type	Value
id	Number	ID
display_name	String	Display name

Get a resource

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	resource	
action	String	Nein	get	
session	String	Nein		
id	String	Nein		The resource id.

Response:

Resource object.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_resource

Search for resources

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	resource	
action	String	Nein	search	
session	String	Nein		
requestBody	JSONObject	Nein		An object with search parameters as described in the table Participant search below.

Response:

An array of resource objects.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Search_for_resources

Participant search

Name Type Value

Search pattern to find resources. In the pattern, the character "*" matches zero or more characters and the character "?" matches exactly one character. All other characters match only themselves.

Module: infostore

Get all infoitems

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	all	
session	String	Nein		
folder	String	Nein		Object ID of the folder, whose contents are queried.
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for infoitems are defined in Common object data and Detailed infoitem data in Anhang B.
sort	Integer	Ja		The identifier of a column which determines the sort order of the response. If this parameter is specified, then the parameter order must be also specified.
order	String	Ja	asc / desc	"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.

Response:

An array with infoitem data. Each array element describes one infoitem and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_all_infoitems

Get a list of infoitems

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	list	
session	String	Nein		
folder	String	Nein		Object ID of the folder, whose contents are queried.
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for infoitems are defined in Common object data and Detailed infoitem data in Anhang B.
objectids	JSONArray	Nein		An array with object IDs of requested infoitems.

Response:

An array with infoitem data. Each array element describes one infoitem and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_list_of_infoitems

Get updated infoitems

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	updates	
session	String	Nein		
folder	String	Nein		Object ID of the folder, whose contents are queried.

columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for infoitems are defined in Common object data and Detailed infoitem data in Anhang B.
sort	Integer	Ja		The identifier of a column which determines the sort order of the response. If this parameter is specified, then the parameter order must be also specified.
order	String	Ja	asc / desc	"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.
timestamp	Long	Nein		Timestamp of the last update of the requested infoitems.
ignore	String	Ja	deleted	Which kinds of updates should be ignored. Currently, the only valid value – "deleted" – causes deleted object IDs not to be returned.

Response:

An array with new, modified and deleted infoitems. New and modified infoitems are represented by arrays. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter. Deleted infoitems are identified by their object IDs as plain strings, without being part of a nested array.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_updated_infoitems

Get an infoitem

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	get	
session	String	Nein		
id	String	Nein		Object ID of the requested infoitem.
version	String	Ja		If present the infoitem data describes the given version. Otherwise the current version is returned

Response:

An object containing all data of the requested infoitem. The field id is not included.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_an_infoitem

Search infoitems**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	search	
session	String	Nein		
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for infoitems are defined in Common object data and Detailed infoitem data in Anhang B.
sort	Integer	Ja		The identifier of a column which determines the sort order of the response. If this parameter is specified, then the parameter order must be also specified.
order	String	Ja	asc / desc	"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.
start	Integer	Ja		The start index (inclusive) in the ordered search, that is requested.
end	Integer	Ja		The last index (inclusive) from the ordered search, that is requested.
requestBody	JSONObject	Nein		JSONObject as described in the table "Search contacts object fields".

Response:

An array with infoitem data. Each array element describes one infoitem and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Search_infoitems

Get an infoitem document

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore/ [filename]	
action	String	Nein	document	
session	String	Nein		
id	String	Nein		Object ID of the requested infoitem.
folder	String	Nein		Object ID of the infoitem's folder.
version	String	Ja		If present the infoitem data describes the given version. Otherwise the current version is returned
content_type	String	Ja		If present the response declares the given content_type in the Content-Type header.

Response:

The raw byte data of the document. The response type for the HTTP Request is set accordingly to the defined mimetype for this infoitem or the content_type given

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_an_infoitem_document

Get all versions

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	versions	
session	String	Nein		
id	String	Nein		Object ID of the infoitem whose versions are requested.
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for infoitems are defined in Common object data and Detailed

				infoitem data in Anhang B.
sort	Integer	Ja		The identifier of a column which determines the sort order of the response. If this parameter is specified, then the parameter order must be also specified.
order	String	Ja	asc / desc	"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.

Response:

An array with infoitem data. Each array element describes one infoitem and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter. The timestamp is the timestamp relating to the requested infostore item.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_all_versions

Update an infoitem

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	update	
session	String	Nein		
id	String	Nein		Object ID of the infoitem whose versions are requested.
timestamp	Long	Nein		Timestamp of the updated infoitem. If the infoitem was modified after the specified timestamp, then the update must fail.
dataobject	JSONObject	Nein		Infoitem object. Only modified fields are present.

Response:

timestamp (Long).

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Update_an_infoitem_via_PUT

Create an infoitem

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	new	
session	String	Nein		
dataobject	JSONObject	Nein		Infoitem object. The field id is not included.

Response:

Object ID of the newly created infoitem.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Create_an_infoitem_via_PUT

Save an attachment in the infostore

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	new	
session	String	Nein		
attached	String	Nein		The Object ID of the Object with the attachment
folder	String	Nein		The Folder ID of the Object with the attachment
module	String	Nein		The Module type of the Object with the attachment.
Attachment	String	Nein		The id of the attachment to save.
dataobject	JSONObject	Nein		Infoitem object as described in Common object data and Detailed infoitem data. The

				field id is not included. The fields in this infoitem object override values from the attachment. The folder_id must be given.
--	--	--	--	--

Response:

Object ID of the newly created infoitem.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Save_an_attachment_in_the_infostore

Delete infoitems

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	delete	
session	String	Nein		
timestamp	Long	Nein		Timestamp of the last update of the deleted infoitems

Response:

An array with objects to delete. The fields for the object are described in the table „Full identifier for an infostore document“ below.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_infoitems

Full identifier for an infostore document

Name	Type	Value
id	Number	Object ID
folder	Number	Folder ID

Delete versions of infostore documents

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	detach	
session	String	Nein		
id	String	Nein		The ID of the base Object
folder	String	Nein		The Folder of the Object
timestamp	Long	Nein		Timestamp of the last update of the deleted infoitems
requestBody	JSONArray	Nein		A List of arrays with the version numbers of the infoitems to detach.

Response:

An array with version numbers that were not deleted.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_versions_of_infostore_documents

Delete all versions of infostore documents

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	revert	
session	String	Nein		
id	String	Nein		The ID of the base Object
folder	String	Nein		The Folder of the Object
timestamp	Long	Nein		Timestamp of the last update of the deleted infoitems

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_all_versions_of_infostore_documents

Copy an infostore document

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	copy	
session	String	Nein		
id	String	Nein		The ID of the base Object
folder	String	Nein		The Folder of the Object
timestamp	Long	Nein		Timestamp of the last update of the deleted infoitems
dataobject	JSONArray	Nein		Infoitem object as described in Common object data and Detailed infoitem data. Only modified fields are present.

Response:

The id of the newly created object.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Copy_an_infostore_document_via_PUT

Lock an infoitem

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	lock	
session	String	Nein		
id	String	Nein		Object ID of the infoitem that should be locked.
diff	Long	Ja		If present the value is added to the current time on the server (both in ms). The document will be locked until that time. If this parameter is not present, the document will be locked for a duration as configured on the server.

Response:

Includes only timestamp and possible errors.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Lock_an_infoitem

Unlock an infoitem

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	infostore	
action	String	Nein	unlock	
session	String	Nein		
id	String	Nein		Object ID of the infoitem that should be locked.

Response:

Includes only timestamp and possible errors.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Unlock_an_infoitem

Modul: Attachments

The Attachment Module allows file attachments to arbitrary objects. Object addresses are defined analogous to the Link module. An Attachment always belongs to an object (called 'attached') in a certain folder of a certain module.

Get all Attachments for an Object

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	attachment	
action	String	Nein	all	
session	String	Nein		
attached	String	Nein		The Object ID of the Object
folder	String	Nein		The Folder ID of the Object
modultype	String	Nein		The Module type of the Object
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for attachment's are defined in Common object data (with only id, created_by and

				creation_date available) and Attachment object.
sort	Integer	Ja		The identifier of a column which determines the sort order of the response. If this parameter is specified, then the parameter order must be also specified.
order	String	Ja		"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.

Response:

An JSONArray with attachment data. Each array element describes one attachment and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_All_Attachments_for_an_Object

Get a list of attachments

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	attachment	
action	String	Nein	list	
session	String	Nein		A session ID previously obtained from the login module
attached	String	Nein		The Object ID of the Object
folder	String	Nein		The Folder ID of the Object
modultype	String	Nein		The Module type of the Object
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for attachment's are defined in Common object data (with only id, created_by and creation_date available) and Attachment object.

dataobject	JSONArray	Nein		An array of with object IDs of requested tasks
------------	-----------	------	--	--

Response with timestamp:

An JSONArray with attachment data. Each array element describes one attachment and is itself an array. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_list_of_attachments

Create an Attachment

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	attachment	
action	String	Nein	attach	
session	String	Nein		A session ID previously obtained from the login module
json_[index]	JSONObject	Nein		The JSON representation of an attachment object as described in Common object data (with only id, created_by and creation_date available) and Attachment Object .
file_[index]	JSONObject	Nein		The file metadata as per <input type=file /> upload.
dataobject	JSONObject	Nein		multipart/form-data or multipart/mixed containing the file data of the attached file and the above fields.

Response:

HTML page with javascript callback as per introduction. Contains a JSONArray of ids of the newly created attachments. The order of the ids corresponds to the indexes in the request.

Attachment Object

ID	Name	Type	Description
800	folder	Number	The ID of the first Folder in which the attached object resides.
801	attached	Number	The object id of the object this attachment is attached to.
802	module	Number	The Module of this Object Possible Values:

			1 Appointment 4 Task 7 Contact 137 Infostore
803	filename	String	The filename of the attached file.
804	file_size	Number	The file size (in bytes) of the attached file.
805	file_mimetype	String	The MIME-Type of the attached file
806	rft_flag	Boolean	If the attachment is a RTF Attachment of Outlook. (Outlook descriptions can be stored as RTF Documents).

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Create_an_Attachment

Delete Attachment

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	attachment	
action	String	Nein	detach	
session	String	Nein		A session ID previously obtained from the login module
attached	String	Nein		The ID of the base Object
modultype	JSONObject	Nein		The type of the Object
dataobject	JSONArray	Nein		An array with the ids of the attachments to delete.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_Attachment

Get updated attachments

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	attachment	
action	String	Nein	updates	
session	String	Nein		A session ID previously obtained from the login module
folder	String	Nein		Object ID of the folder, whose contents are queried
attached	String	Nein		The ID of the base Object
moduleid	String	Nein		Module ID (as per Attachment Object) of the attached object.
columns	String	Nein		A comma-separated list of columns to return. Each column is specified by a numeric column identifier. Column identifiers for attachments are defined in Common object data (with only id, created_by and creation_date available) and Attachment Object .
sort	Integer	Ja		The identifier of a column which determines the sort order of the response. If this parameter is specified, then the parameter order must be also specified.
order	String	Ja		"asc" if the response entires should be sorted in the ascending order, "desc" if the response entries should be sorted in the descending order. If this parameter is specified, then the parameter sort must be also specified.
timestamp	Time	Nein		Timestamp of the last update of the requested infoitems.
ignore	String	Ja		Which kinds of updates should be ignored. Currently, the only valid value – "deleted" – causes deleted object IDs not to be returned.

Response with timestamp:

An array with new and deleted attachments for the specified object. New attachments are represented by arrays. The elements of each array contain the information specified by the corresponding identifiers in the columns parameter. Deleted attachments are identified by their object IDs as plain numbers, without being part of a nested array.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_updated_attachments

Get an Attachment

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	attachment	
action	String	Nein	get	
session	String	Nein		A session ID previously obtained from the login module
folder	String	Nein		Object ID of the folder, whose contents are queried
attached	String	Nein		Object ID of the object to which the attachments are attached.
moduleid	String	Nein		Module ID (as per Attachment Object) of the attached object.
id	String	Nein		Object ID of the requested attachment.

Response with timestamp:

An JSONObject containing all data of the requested attachment. The fields of the object are listed in Common object data(with only id, created_by and creation_date available) and Attachment Object.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_an_attachment

Get an attachments filedata**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	Attachment [filename]	
action	String	Nein	document	
session	String	Nein		A session ID previously obtained from the login module
folder	String	Nein		Object ID of the folder, whose contents are queried
attached	String	Nein		Object ID of the object to which the attachments are attached.

moduleid	String	Nein		Module ID (as per Attachment ObjectAttachment Object) of the attached object.
id	String	Nein		Object ID of the requested attachment.
content_type	String	Ja		If set the responses Content-Type header is set to this value, not the attachments file mime type.

Response:

The raw byte data of the document. The response type for the HTTP Request is set accordingly to the defined mimetype for this infoitem. Note: The Filename may be added to the customary infostore path to suggest a filename to a Save-As dialog.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_an_attachments_filedata

Modul: reminder

The reminder module provides the ability to fetch all active reminders for a user between two dates.

Get a reminder range

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	reminder	
action	String	Nein	range	
session	String	Nein		
end	String	Nein		The End date of the reminder range

Response:

An JSONArray with all reminders which are scheduled until the specified time. Each reminder is described in Reminder response.

Reminder response

Field	Type	Decription
id	Number	The ID of the reminder.
target_id	Number	The target_id where this reminder is attached to
alarm	Time	The time of the alarm
module	Number	The module of the reminder
servertime	Time	The time on the server
user_id	Number	The ID of the user.
last_modified	Time	The last modification timestamp of the reminder
recurrence_position	Time	The recurrence position for series appointments or 0 if no series
folder	Number	The ID of the folder through that the object can be read

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_reminder_range

Delete Reminder

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	reminder	
action	String	Nein	range	
session	String	Nein		
dataobject	JSONObject	Nein		An object with the field "id".

Response:

An JSON array with the id that was not deleted.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_Reminder

Modul: multiple

The multiple module allows to bundle multiple requests to most other modules in a single request. Not supported are:

- modules login, config and multiple,
- POST requests with a multipart encoding (uploads),
- GET requests which do not use an object as described in Response body as response body (downloads).

Multiple requests

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	multiple	
session	String	Nein		
continue		Nein		Specifies whether processing of requests should stop when an error occurs, or whether all request should be processed regardless of errors.
dataobject	JSONArray	Nein		An array with request objects. Each request object contains all URI parameters of the "normal" request as fields. The module name of the "normal" request is included in the field module. The parameter session is not included. If the "normal" request has a request body, the object which is represented by that request body is includes as the value of the field data.

Response:

An array with reply objects as described in Response body. The order of reply objects corresponds to the order of requests in the request body. Unlike with all other modules, this response is itself not part of a response object as described in Response body.

Response body

Name	Type	Value
data	Value	Payload of the response.
timestamp	Timestamp	Current timestamp on the server.

error	String	English error message. Present in case of errors.
error_params	Array	Replacement parameters for the error message
error_id	String	Unique error identifier to help finding this error instance in the server logs.
code	String	Error code consisting of a three-letter category and a four-digit message number, separated by a dash.
category	Number	<p>Category to which the error message belongs:</p> <ul style="list-style-type: none"> 1 An error resulting from wrong or missing input from front-end (e.g. mandatory field missing). 2 An error strictly related to user configuration which denies requested operation. 3 An error related to insufficient permission settings. 4 A requested operation could not be accomplished because a needed resource is temporary down or missing (e.g. imap server rejects connection because of too many established connections). 5 A subsystem or third party service is down and therefore does not respond (e.g. database is down). 6 The underlying socket connection is corrupt, empty or closed. Only a temporary error that does not affect the whole system. 7 An internal java-related (runtime) exception. 8 A programming error which was caused by incorrect program code. 9 A concurrent modification. 10 Error in system setup detected. 11 The requested operation could not be performed cause an underlying resource is full or busy (e.g. IMAP folder exceeds quota). 12 The given data could not be stored into the database because an attribute contains a too long value. 13 Action was at least partially successful, but a condition occurred that merited a warning

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Multiple_requests

Modul: quota

Dieses Modul stellt die Funktionalität zur Verfügung um die Informationen über die Benutzung des Dateispeichers abzufragen.

Get the filestore usage data

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	quota	
action	String	Nein	filestore	
session	String	Nein		

Response:

JSONObject mit den Felder „use“ und „quota“

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_the_filestore_usage_data

Get the mail usage data

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	quota	
action	String	Nein	mail	
session	String	Nein		

Response:

A JSON Object containing the fields “use” and “quota”. “use” represents the use mail quota and the field “quota” represents the maximum. -1 represents an unlimited quota.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_the_mail_usage_data

Modul: import

Dieses Modul stellt die Funktionalität zur Verfügung, um die Daten für die anderen Modulen (wie Contacts, Tasks oder Appointments) in verschiedenen Formaten (iCal, vCard, CSV) ins Ordner zu importieren.

Import CSV

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	import	
action	String	Nein	CSV	
session	String	Nein		
folder	String	Nein		ObjectID des Ordners, wohin die Inhalte importiert werden müssen. Es muss ein Ordner mit dem Inhalt vom ContentType Contacts sein
dataobject	.CSV file	Nein		.CSV Datei als "multipart/form-data" kodiert. Die Spaltentitel sind wie in OX. Siehe die Spalte DisplayedNames in #DetailedContactData

Response:

JSONArray von JSONObject, ein JSONObject für jeden Eintrag in der Liste. Ein JSONObject enthält die ObjectId des Eintrages, die Id des Ordners, Timestamp der Modifikation und im Falle eines Fehlers eine Fehlermeldung.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Import_CSV

Import Outlook CSV

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	import	
action	String	Nein	OUTLOOK_CSV	
session	String	Nein		

folder	String	Nein		ObjectID des Ordners, wohin die Inhalte importiert werden müssen. Es muss ein Ordner mit dem Inhalt vom ContentType Contacts sein
dataobject	.CSV file	Nein		.CSV Datei als Windows-1252 kodiert. Die Spaltentitel sind wie bei der englische, französische oder deutsche Version vom Outlook

Response:

JSONArray von JSONObject, ein JSONObject für jeden Eintrag in der Liste. Ein JSONObject enthält die ObjectID des Eintrages, die Id des Ordners, Timestamp der Modifikation und im Falle eines Misserfolges beim Import eine Fehlermeldung.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Import_Outlook_CSV

Import iCAL

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	import	
action	String	Nein	iCAL	
session	String	Nein		
folder	JSONArray	Nein		ObjectIDs der Ordner, wohin die Inhalte importiert werden müssen. Es müssen Ordner mit dem Inhalt entweder ContentType Tasks oder Appointment sein. Es ist auch eine Liste mit Ordner von beiden Typen erlaubt.
dataobject	iCalendar file	Nein		

Response:

JSONArray von JSONObject, ein JSONObject für jeden Eintrag in der Liste. Ein JSONObject enthält die ObjectID des Eintrages, die Id des Ordners, Timestamp der Modifikation, im Falle einer Fehlers eine Fehlermeldung und Warnungen (unter dem Schlüssel „warnings“). Die Warnungen enthalten ein Array von Feldern mit Errors.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Import_iCAL

Import vCard

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	import	
action	String	Nein	VCARD	
session	String	Nein		
folder	String	Nein		ObjectID des Ordners, wohin die Inhalte importiert werden muss. Es muss ein Ordner mit dem Inhalt vom ContentType Contacts sein
dataobject	vCard file	Nein		Mögliche Formaten: vCard 2.1, vCard 3.0 or vCalendar 1.0

Response:

JSONArray von JSONObject, ein JSONObject für jeden Eintrag in der Liste. Ein JSONObject enthält die ObjectId des Eintrages, die Id des Ordners, Timestamp der Modifikation und im Falle eines Misserfolges beim Import eine Fehlermeldung.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Import_vCard

Modul: export

Dieses Modul stellt die Funktionalität zur Verfügung, um die Daten für die anderen Modulen (wie Contacts, Tasks oder Appointments) in verschiedenen Formaten (iCal, vCard, CSV) aus dem Folder zu exportieren.

Exporting CSV

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	export	
action	String	Nein	CSV	
session	String	Nein		
folder	String	Nein		ObjectID des Ordners, dessen Inhalt exportiert werden muss. Es muss ein Ordner mit dem Inhalt vom ContentType Contacts sein
columns	Array	Ja		Die Spalten, die von der gegebenen Datei exportiert werden müssen, als Array von Spaltennummern.

Response:

InputStream mit der Datei vom Typ ‚MIME type text/csv‘

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Exporting_CSV

Exporting iCAL

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	export	
action	String	Nein	ICAL	
session	String	Nein		
folder	String	Nein		ObjectID des Ordners, dessen Inhalt exportiert werden muss. Es muss ein Folder mit dem Inhalt vomContentType Calendar sein

Response:

InputStream mit der Datei vom Typ MIME type text/calendar

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Exporting_iCAL

Exporting vCAD

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	export	
action	String	Nein	VCARD	
session	String	Nein		
folder	String	Nein		ObjectID des Ordners, dessen Inhalt exportiert werden muss. Es muss ein Ordner mit dem Inhalt vom ContentType Contacts sein

Response:

InputStream mit der Datei vom Typ ‚MIME type text/x-vcard‘

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Exporting_vCard

Modul: sync

Dieses Modul liefert einige API Erweiterungen, um die Operationen in den mobilen Synchronisierungsumgebungen zu unterstützen.

Clearing a folder's content

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	sync	
action	String	Nein	refresh_server	
session	String	Nein		
dataobject	JSONArray	Nein		enthält IDs der Ordner, deren Inhalt gelöscht werden muss

Response:

JSONArray mit ids der Ordner, die wegen Konflikten durch Nebenläufigkeit nicht gelöscht wurden

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Clearing_a_folder.27s_content

Modul: mailfilter

Dieses Modul beschreibt wie man die Regeln zum Filtern der Emails hinzufügen, löschen oder updaten kann. Außerdem bietet dieses Modul die Möglichkeit zu prüfen, welche Aktionen bei dem zu Grunde liegenden System unterstützt werden.

Detaillierte Beschreibung dieses Moduls befindet sich unter dem Link:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API_MailFilter

Modul: ajax file upload

Dieses Modul ermöglicht die Speicherung der Dateien in bestimmten Verzeichnissen beim Server für konfigurierbaren Zeitperioden. Diese Dateien sind dann zugänglich für weitere Operationen genauso wie inline-Bilder in (html)Mails.

Uploading a file

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	file	

action	String	Nein	new	
session	String	Nein		
type	String	Nein		Der Filter für Dateien-Typen um zu bestimmen, welche Dateien für Upload erlaubt sind. Zur Zeit unterstützte Filter sind: <i>file=all, text=text/*, media=image OR audio OR video, image=image/*, audio=audio/*, video=video/*, application=application/*</i>
dataobject	JSONObject	Nein		Allgemeine Post RequestBody vom MIME Typ "multipart/*", welches die Datei für Upload enthält

Response:

JSONArray mit ids der hochgeladenen Dateien

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Uploading_a_file

Uploading a file's last access timestamp (keep alive)

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	file	
action	String	Nein	keepalive	
session	String	Nein		
id	String	Nein		die ID der hochgeladenen Datei, deren Zeitstempel aktualisiert werden muss

Response:

Der String mit dem Wert „null“ als ‚data‘

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Updating_a_file.27s_last_access_timestamp_.28keep_alive.29

Requesting a formerly uploaded a file

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	file	
action	String	Nein	get	

session	String	Nein		
id	String	Nein		die ID der hochgeladenen Datei

Response:

Der Inhalt der abgefragten Datei, der direkt in den OutputStream geschrieben wird

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Requesting_a_formerly_uploaded_file

Modul: image

Dieses Modul erlaubt das Herunterladen der Bilder vom Open-Xchange Server ohne Sessionid in Request anzugeben.

Requesting an image

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
uid	String	Nein		Eindeutige ID für das gewünschte Bild

Response:

Der Inhalt des gelieferten Bildes, der direkt in OutputStream geschrieben wird

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Requesting_an_image

Modul: coversion (preliminary)

A generic module to request data from a data source and to process obtained/submitted data with a data handler. Thus data is converted from a data source by a data handler.

Converting data

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	conversion	
action	String	Nein	convert	

dataobject	JSONObject	Nein		A Conversion request JSON object containing nested JSON objects for Data source object and Data handler object
------------	------------	------	--	---

Conversion request JSON object

Field	Type	Description
datasource	JSON object	The data source object.
datahandler	JSON object	The data handler object.

Data source object

Field	Type	Description
identifier	String	The identifier of the data source.
args	JSON array	The optional name-value-pairs as JSON objects for data source

Data handler object

Field	Type	Description
identifier	String	The identifier of the data handler.
args	JSON array	The optional name-value-pairs as JSON objects for data handler

Response:

The result of converted data ready as an appropriate JSON response

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Converting_data

Saving an ICal email attachment

If an ICal file is attached to an email, its content can be saved as appointments and tasks into given calendar and task folder.

Data source's JSON object

```
{
  "identifier": "com.openexchange.mail.ical"
  "args":
    [
      {"com.openexchange.mail.conversion.fullname": "<folder-fullname>"}
      {"com.openexchange.mail.conversion.mailid": "<mail-id>"}
      {"com.openexchange.mail.conversion.sequenceid": "<attachment-
sequence-id>"}
    ]
}
```

Data handler's JSON object

```
{
  "identifier": "com.openexchange.ical"
  "args":
    [
      {"com.openexchange.groupware.calendar.folder": "<calendar-folder-id>"}
      {"com.openexchange.groupware.task.folder": "<task-folder-id>"}
    ]
}
```

Response: A JSON array of JSON objects each providing folder and object ID of added appointment/task; e.g. [{"folder_id":2567, "id":7689}, ...]

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Saving_an_ICal_email_attachment

Saving a vCard email attachment

If a VCard file is attached to an email, its content can be saved as contacts into given contact folder.

Data source's JSON object

```
{
  "identifier": "com.openexchange.mail.vcard"
  "args":
    [
      {"com.openexchange.mail.conversion.fullname": "<folder-fullname>"}
      {"com.openexchange.mail.conversion.mailid": "<mail-id>"}
      {"com.openexchange.mail.conversion.sequenceid": "<attachment-sequence-id>"}
    ]
}
```

Data handler's JSON object

```
{
  "identifier": "com.openexchange.contact"
  "args":
    [
      {"com.openexchange.groupware.contact.folder": "<contact-folder-id>"}
    ]
}
```

Response:

A JSON array of JSON objects each providing folder and object ID of added contact; e.g. [{"folder_id":2567, "id":7689}, ...]

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Saving_a_VCard_email_attachment

Contact(s) attached to a new email as a VCard file

Obtain VCard data from specified contact object(s).

Data source's JSON object

```
{
```

```

"identifier": "com.openexchange.contact"
"args":
[
  {"folder": "<folder-id1>", "id": "<id1>"}
  ...
  {"folder": "<folder-idn>", "id": "<idn>"}
]
}

```

Get a new email's JSON object with specified VCard data source attached.

Data handler's JSON object

```

{
  "identifier": "com.openexchange.mail.vcard"
  "args": []
}

```

Response:

A mail JSON object.

OX Dokumentation:

http://www.openexchange.com/wiki/index.php?title=HTTP_API#Contact.28s.29_attached_to_a_new_email_as_a_VCard_file

Modul: mail account (preliminary)

Dieses Modul dient zum Verwalten mehrfachen Email Accounts.

Get a mail account

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	account	
action	String	Nein	get	
session	String	Nein		
id	String	Nein		die ID des Kontos, das zurückgeliefert werden muss

Response:

JSONObject mit den Feldern, die in folgender Tabelle Mail account data aufgelistet sind:

ID	Name	Type	Value
----	------	------	-------

1001	id	Number	Account ID
1002	login	String	The login.
1003	password	String	The (optional) password.
1004	mail_url	String	The mail server URL; e.g. "imap://imap.somewhere.com:143". URL is preferred over single fields (like mail_server, mail_port, etc.)
1005	transport_url	String	The transport server URL; e.g. "smtp://smtp.somewhere.com:25". URL is preferred over single fields (like transport_server, transport_port, etc.)
1006	name	String	Account's display name.
1007	primary_address	String	User's primary address in account e.g. "someone@somewhere.com"
1008	spam_handler	String	The name of the spam handler used by account.
1009	trash	String	The name of the default trash folder.
1010	sent	String	The name of the default sent folder.
1011	drafts	String	The name of the default drafts folder.
1012	spam	String	The name of the default spam folder.
1013	confirmed_spam	String	The name of the default confirmed spam folder.
1014	confirmed_ham	String	The name of the default confirmed ham folder.
1015	mail_server	String	The mail server's hostname or IP address.
1016	mail_port	Number	The mail server's port.
1017	mail_protocol	String	The mail server's protocol. Always use basic protocol name. E.g. use "imap" instead of "imaps"
1018	mail_secure	Boolean	Whether to establish a secure connection to mail server (SSL, TLS)
1019	transport_server	String	The transport server's hostname or IP address.
1020	transport_port	Number	The transport server's port.
1021	transport_protocol	String	The transport server's protocol. Always use basic protocol name. E.g. use "smtp" instead of "smtps"
1022	transport_secure	Boolean	Whether to establish a secure connection to transport server (SSL, TLS).
1023	transport_login	String	The transport login. If empty "login" is taken as transport login as well.
1024	transport_password	String	The transport password. If empty "password" is taken as transport password as well.
1025	unified_inbox_enabled	Boolean	If enabled for Unified INBOX
1026	trash_fullname	String	Path to default trash folder. Preferred over "trash"
1027	sent_fullname	String	Path to default sent folder. Preferred over "sent"

1028	drafts_fullname	String	Path to default drafts folder. Preferred over "drafts"
1029	spam_fullname	String	Path to default spam folder. Preferred over "spam"
1030	confirmed_spam_fullname	String	Path to default confirmed-spam folder. Preferred over "confirmed spam"
1031	confirmed_ham_fullname	String	Path to default confirmed-ham folder. Preferred over "confirmed ham"
1032	pop3_refresh_rate	Number	The interval in minutes the POP3 account is refreshed
1033	pop3_expunge_on_quit	Boolean	Whether POP3 messages shall be deleted on actual POP3 account after retrieval or not
1034	pop3_delete_write_through	Boolean	If option "pop3_expunge_on_quit" is disabled, this property defines whether a delete in local INBOX also deletes affected message in actual POP3 account
1035	pop3_storage	String	The name of POP3 storage provider. default is "mailaccount"
1036	pop3_path	String	Path to POP3's virtual root folder in storage, default is name of the POP3 account beside default folder

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Get_a_mail_account

Create a new mail account

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	account	
action	String	Nein	new	
session	String	Nein		
dataobject	JSONObject	Nein		Ein Konto mit den Feldern, die in der Tabelle aufgelistet sind

Response:

JSONObject mit den Feldern, die in der Tabelle aufgelistet sind

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Create_a_new_mail_account

Update a mail account

Parameter:

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	account	
action	String	Nein	update	
session	String	Nein		
dataobject	JSONObject	Nein		Ein Konto mit den Feldern, die in der Tabelle aufgelistet sind

Response:

JSONObject mit den Feldern, die in der Tabelle aufgelistet sind

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Update_a_mail_account

Delete a mail account**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	account	
action	String	Nein	delete	
session	String	Nein		
ids	Array	Nein		Ein Array mit ids der Mail-Kontos, die gelöscht werden müssen

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Delete_a_mail_account

Validate a mail account (which shall be created)**Parameter:**

Name	Datentyp	Optional	Wert	Bedeutung
module	String	Nein	account	
action	String	Nein	validate	
session	String	Nein		
dataobject	JSONObject	Nein		Ein Konto zum Validieren.

				Siehe .
tree	Boolean	Ja		Dieser Parameter zeigt, ob beim Erfolg ein Folder-Baum zurückgeliefert werden muss (NULL im Falle des Misserfolges). Wenn der Parameter nicht gesetzt ist oder auf false gesetzt ist, dann wird als Ergebnis nur Boolean - Wert zurückgeliefert.

Response:

Abhängig vom Setzen des Parameters tree wird entweder Folder als JSONObject oder Boolean -Wert zurückgeliefert.

OX Dokumentation:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API#Validate_a_mail_account_.28which_shall_be_created.29

Anhang A: Liste der USM Error Codes

Configuration Error Codes:

0x130000 to 0x130004 - Configuration property not set

0x130010 - Properties file not found.

0x130020 - Error while reading USM config file

0x131000 to 0x131003 - Missing configuration property

0x131012 to 0x131013 - Invalid value for configuration property

Content Type Error Codes:

0x400000 - Client is requesting fields from server which are not supported

0x400001 - Clients is requesting fields from server which not exist

0x400002 - Client is receiving objects with invalid content type

0x400003 - Client is sending objects with invalid content type

0x400010 to 0x400019 - OX send invalid/unexpected JSON result to the client

0x400020 to 0x400024 - Data array is not present where expected in a JSON Result received from the OX server

0x400030 to 0x400031 - OX server sent invalid timestamp in the JSON result

0x400040 to 0x400044 - OX server did not sent ID for a new object

0x400050 to 0x400054 - Error when reading the data from the JSON Result received form OX

0x400060 to 0x400064 - Error when writing the data from the JSON Result received form OX to an USM object or when writing data to a JSON object

0x400070 - Error when deleting object from server

0x400080 to 0x400081 - Error when confirming task or appointment

Calendar:

0x460000 - Conflicts occurred when creating appointments (time and resources conflicts)

0x460001 - Appointment could not be created because of ox communication error

Contacts:

0x440000 to 0x440001 - Error when reading /writing image from/to the server

Folder:

0x420000 - Error when trying to compose a hierarchy tree with invalid nodes

Mail:

0x430000 - Client is trying to create new email object on the server

0x430001 - Required field is missing

Manager:

0x410000 - Content Type already exists

0x410001 - Not enough fields in result; at least ID and parent folder ID are required

0x410002 - Field 0 is not ID field

0x410003 - Field 1 is no parent folder ID field

Database:

Hsql:

0x210000 - Couldn't load Hsql JDBC driver
0x210001 - Couldn't connect to HsqlDB
0x210002 - Error during shutdown of HsqlDB
0x210003 - Error while closing HsqlDB
0x210004 - Couldn't connect to HsqlDB
0x210005 - Error while closing HsqlDB connection
0x210006 - Couldn't read connection properties for HsqlDB
0x210007 - Couldn't create sequence
0x210008 - Couldn't create sequence because of autocommit settings
0x210009 to 0x21000A - Error when getting next value from sequence
0x21000B - Getting new value from sequence returned no data

MySql:

0x220000 - Couldn't load MySql JDBC driver
0x220001 - Couldn't connect to MySql DB
0x220002 - Error while closing MySql DB
0x220003 - Couldn't read connection properties for MySql
0x220004 - Couldn't create sequence
0x220005 - Couldn't create sequence because of autocommit settings
0x220006 - Error when getting next value from sequence

OX:

0x230000 - Couldn't get read only connection
0x230001 - Couldn't get read/write connection
0x230002 - Couldn't create sequence
0x230003 - Couldn't create sequence because of autocommit settings

0x230004 - Error when getting next value from sequence
0x230005 - Getting new value from sequence returned no data

Mapping:

0x610001 - Mapping update in DB failed
0x610002 - Reading cache from DB failed
0x610003 - Generating Short ID failed
0x610004 - SQL error retrieving the UID for an object
0x610005 - SQL error storing the UID for an object

USM Session Management:

0x300000 - Can not serialize data to byte array
0x300001 - Configuration for Sync retries is invalid
0x300002 to 0x300005 - Illegal change state of an object
0x300006 to 0x300005 - Illegal Conflict Resolution
0x300008 to 0x30000B - Error storing cache data
0x30000C to 0x30000E - SQL error while retrieving / writing persistent data from DB
0x300010 - Too many repeated conflicts on slow sync
0x300011 - Conflicting changes between server and client on slow sync
0x300012 - Too many repeated conflicts on incremental sync
0x300013 - Conflicting changes between server and client on incremental sync
0x300014 - No cached DataObjects for client available
0x300015 to 0x300017 - Invalid configuration value
0x300018 - Stored object is not of correct type
0x300019 - Can not retrieve object from bytes
0x30001A - Can not find the class of the cached object
0x30001B - USM access disabled for user

0x30001C - Configuration for max_states_in_db must be greater or equal to 2

0x30001D - Configuration for max_states_in_db is invalid

0x30001E - Configuration for keep_last_state_in_db is invalid

0x30001F - Configuration for usm interface module is invalid

JSON Error Codes:

0x180000 to 0x180002 - Unsupported methods in AbstractJSONResult.

0x180003 - Unknown OX response when reading configuration

0x180004 - OX configuration response contains illegal data

0x180005 - Authentication failed when trying to login.

0x180006 - Server response did not contain session ID

0x180007 - Unsupported Encoding: Character encoding UTF-8 not found

0x180008 - OXJSONAccess not activated

0x18000A - Character encoding UTF-8 not found

0x18000B - HTTP Connection Error

0x18000C - Conflicting change on OX access

0x18000D - HTTP Exception occurred

0x18000E - IO Exception occurred

0x18000F - Invalid Status Code when retrieving a resource

0x180010 - Invalid URL specified when retrieving a resource

0x180011 - HTTP Exception occurred when retrieving a resource

0x180012 - IO Exception occurred when retrieving a resource

0x180013 - Invalid Status Code when storing a resource

0x180014 - Invalid URL specified when storing a resource

0x180015 - HTTP Exception occurred when retrieving a resource

0x180016 - IO Exception occurred when retrieving a resource

0x180017 - Result of POST request doesn't match configured pattern

0x180018 - Result of POST request doesn't contain valid JSONObject

0x180019 - Result of POST request doesn't contain expected action

0x180020 - Couldn't read OX server access properties

Toolkit (Conversion errors):

0x110000 to 0x110001 - UTF-8 not supported

Anhang B USM ContentTypes

Dieser Anhang enthält die Beschreibung allen in USM definierten ContentTypes.

Die in USM definierten ContentTypes unterscheiden sich nur in wenigen Punkten von der OX-ContentTypes. Die Beschreibung von OX-ContentTypes befindet sich unter dem Link:

http://www.open-xchange.com/wiki/index.php?title=HTTP_API

Jeder ContentType hat immer ein Feld namens „uuid“. In diesem wird der eindeutige Identifier für dieses Objekt gespeichert. Immer wenn ein Objekt übergeben oder referenziert wird, muss dieser Identifier gesetzt sein.

Jedes vom USM gemeldete DataObject hat immer das Feld „objectType“ gesetzt, in ihm wird der Identifier des ContentTypes des DataObjects übergeben. Er entspricht immer dem Feld „module“ des enthaltenden Folders. Ein Client darf ebenfalls dieses Feld senden (muss es aber nicht), es muss ebenfalls genau dem Feld „module“ des enthaltenden Folders entsprechen. Eine Veränderung des „type“-Feldes ist nicht zulässig.

FolderContentType

Identifikator

folder

ContentType Daten

Im Falle FolderContentType setzen sich die Daten aus *Common folder data* und *Detailed folder data* zusammen.

Common folder data

id	Name	Type	Value
1	id	String	Object ID
2	created_by	String	User ID of the user who created the object.
3	modified_by	String	User ID of the user who last modified this object.
4	creation_date	Time	Date and time of creation.
5	last_modified	Time	Date and time of the last modification.

6	last_modified_utc	Timestamp	Timestamp of the last modification. Note that the type is Timestamp, not Time. See #Date and time for details.
20	folder_id	String	Object ID of the parent folder.
	folder_uuid	String	UUID of the parent folder. Will be reported if "folder_id" is reported (i.e. on changes to folder_id also folder_uuid is reported). The client can modify/set either folder_uuid or folder_id, folder_uuid has a higher priority (overwrites the value of folder_id) if both fields are set by the client

Detailed folder data

id	Name	Type	Value
300	title	String	Name of this folder.
301	module	String	Name of the module which implements this folder; e.g. "tasks", "calendar", "contacts", "infostore", or "mail"
302	type	Number	Type of folder: 1 private 2 public 3 shared 5 system folder
304	subfolders	Boolean	true if this folder has subfolders.
305	own_rights	Number	Permissions which apply to the current user, as described either in Permission flags or in RFC 2086.
306	permissions	Array	Each element is an object described
307	summary	String	Information about contained objects.
308	standard_folder	Boolean	Indicates whether or not folder is marked as a default folder (only OX folder)
309	total	Number	The number of objects in this Folder
310	new	Number	The number of new objects in this Folder.
311	unread	Number	The number of unread objects in this

			Folder.
312	deleted	Number	The number of deleted objects in this Folder.
313	capabilities	Number	Bit mask containing information about mail folder capabilities, as described in Capabilities.
314	subscribed	Boolean	Indicates whether this folder should appear in folder tree or not.
315	subscr_subflds	Boolean	Indicates whether subfolders should appear in folder tree or not.

Permission flags

Bits	Value
0-6	Folder permissions: 0 No permissions. 1 See the folder. 2 Create objects in the folder. 4 Create subfolders. 64 Admin.
7-13	Read permissions for objects in the folder: 0 No permissions. 1 Read only own objects. 2 Read all objects. 64 Admin.
14-20	Write permissions for objects in the folder: 0 No permissions. 1 Modify only own objects. 2 Modify all objects. 64 Admin.
21-27	Delete permissions for objects in the folder: 0 No permissions. 1 Delete only own objects. 2 Delete all objects. 64 Admin.
28	Admin flag: 0 No permissions. 1 Allowed to modify permissions.

Permission object

Name	Type	Value
bits	Number	For non-mail folders, a number as described in Permission flags
rights	String	For mail folders, the rights string as defined in RFC 2086.

entity	Number	User ID of the user or group to which this permission applies.
group	Boolean	true if entity refers to a group, false if it refers to a user.

Capabilities

Bit	Description
0	Mailing system supports permissions.
1	Mailing system supports ordering mails by their thread reference.
2	Mailing system supports quota restrictions.
3	Mailing system supports sorting.
4	Mailing system supports folder subscription.

Common object data

Common object data stellen die Eigenschaften, die einige ContentTypes gemeinsam haben, dar. Sie werden in diesem Abschnitt in einer separaten Tabelle dargestellt. Wenn ein ContentType diese Daten enthält, wird im Folgenden bei der Beschreibung des entsprechenden ContentTypes auf diese Tabelle hingewiesen.

Common object data

ID	Name	Type	Value
1	id	String	Object ID
2	created_by	String	User ID of the user who created this object.
3	modified_by	String	User ID of the user who last modified this object.
4	creation_date	Time	Date and time of creation.
5	last_modified	Time	Date and time of the last modification.
20	folder_id	String	Object ID of the parent folder.
100	categories	String	String containing comma separated the categories. Order is preserved. Changing the order counts as

			modification of the object. Not present in folder objects.
101	private_flag	Boolean	Overrides folder permissions in shared private folders: When true, this object is not visible to anyone except the owner. Not present in folder objects.
102	color_label	Number	Color number used by Outlook to label the object. The assignment of colors to numbers is arbitrary and specified by the client. The numbers are integer numbers between 1 and 10 (inclusive). Not present in folder objects.
104	number_of_attachments	Number	Number of attachments

TaskContentType

Identifikator

tasks

ContentType Daten

Im Falle TaskContentType setzen sich die Daten aus Common object data, *Detailed task and appointment data* und *Detailed task data* zusammen.

Detailed task and appointment data

ID	Name	Type	Value
200	title	String	Short description.
201	start_date	Time	Inclusive start of the event as Date for tasks and whole day appointments and Time for normal appointments. For sequencies, this date must be part of the sequence, i. e. sequencies always start at this date.
202	end_date	Time	Exclusive end of the event as Date for tasks and whole day appointments and as Time for normal appointments.
203	note	String	Long description.
204	alarm	Time	Specifies when to notify the participants as the number of minutes before the start of the

			appointment. For tasks, the Time value specifies the absolute time when the user should be notified.
207	recurrence_position	Number	1-based position of an individual appointment in a sequence. Present if and only if recurrence_type > 0.
208	recurrence_date_position	Date	Date of an individual appointment in a sequence. Present if and only if recurrence_type > 0.
209	recurrence_type	Number	Specifies the type of the recurrence for a task sequence: 0 none (single event) 1 daily 2 weekly 3 monthly 4 yearly
212	days	Number	Specifies which days of the week are part of a sequence. The value is a bitfield with bit 0 indicating sunday, bit 1 indicating monday and so on. May be present if recurrence_type > 1. If allowed but not present, the value defaults to 127 (all 7 days).
213	day_in_month	Number	Specifies which day of a month is part of the sequence. Counting starts with 1. If the field "days" is also present, only days selected by that field are counted. If the number is bigger than the number of available days, the last available day is selected. Present if and only if recurrence_type > 2.
214	month	Number	Month of the year in yearly sequences. 0 represents January, 1 represents February and so on. Present if and only if recurrence_type = 4.
215	interval	Number	Specifies an integer multiplier to the interval specified by recurrence_type. Present if and

			only if recurrence_type > 0. Must be 1 if recurrence_type = 4.
216	until	Date	Inclusive end date of a sequence. May be present only if recurrence_type > 0. The sequence has no end date if recurrence_type > 0 and this field is not present. Note: since this is a Date, the entire day after the midnight specified by the value is included.
217	notification	Boolean	If true, all participants are notified of any changes to this object. This flag is valid for the current change only, i. e. it is not stored in the database and is never sent by the server to the client.
220	participants	Array	Each element identifies a participant, user group or booked resource as described in
221	users	Array	Each element represents a participant as described in User participant object. User groups are resolved and are represented by their members. Any user can occur only once.
222	occurrences	Number	Specifies how often a recurrence should appear. May be present only if recurrence_type > 0.

Participant identifier

Name	Type	Value
id	Number	User ID
type	Number	Type of participant: 1 user 2 user group 3 resource 4 resource group 5 external user

User participant object

Name	Type	Value
------	------	-------

id	Number	User ID
display_name	String	Displayable name of the participant.
confirmation	Number	0 none 1 accepted 2 declined 3 tentative
confirmmessage	String	Confirm Message of the participant

Detailed task data

ID	Name	Type	Value
300	status	Number	Status of the task: 1 not started 2 in progress 3 done 4 waiting 5 deferred
301	percent_completed	Number	How much of the task is completed. An integer number between 0 and 100.
302	actual_costs	Number	
303	actual_duration	Number	
305	billing_information	Number	
306	project_id	Number	
307	target_costs	Number	
308	target_duration	Number	
309	priority	Number	1 = LOW, 2 = MEDIUM, 3 = HIGH
312	currency	Number	
313	trip_meter	Number	
314	companies	Number	
315	date_completed	Number	

AppointmentContentType

Identifikator

calendar

ContentType Daten

Im Falle AppointmentContentType setzen sich die Daten aus **Common object data**, Detailed task and appointment data *und Detailed appointment data* zusammen.

Detailed appointment data

ID	Name	Type	Value
206	recurrence_id	Number	Object ID of the entire appointment sequence. Present on series and change exception appointments. Equals to object identifier on series appointment and is different to object identifier on change exceptions.
400	location	String	Location
401	full_time	Boolean	True if the appointment is a whole day appointment, false otherwise.
402	shown_as	Number	Describes, how this appointment appears in availability queries: 1 reserved 2 temporary 3 absent 4 free

Im Falle Serientermine gibt es zwei zusätzliche Felder, die gesetzt werden können, um eventuelle Ausnahmen zu bestimmen: „change_exceptions“ und „delete_exceptions“.

„**delete_exceptions**“ ist eine JSONArray, die die Startzeiten der gelöschten Termine der Serie enthält.

Ausnahmen als Löschungen kann man bei der Generierung einer Serie mitschicken, z.B.:

```
"created": [{"interval": 1, "color_label": 0, "created_by": "7", "end_date": 1255684700000, "folder_id": "41", "full_time": false, "id": "2465", "modified_by": "7", "note": "Do not delete !", "objectType": "calendar", "participants": [{"id": 7, "type": 1}], "private_flag": false, "recurrence_type": 1, "shown_as": 1, "start_date": 1255683700000, "title": "Serie", "users": [{"confirmation": 1, "confirmmessage": "", "display_name": "", "id": 7}], "delete_exceptions": [1255676500000, 1256803300000], "uuid": "05f46699-5a78-4b28-9747-3f755f47eb2b"}]
```

oder als Änderung der Serie schicken:

```
"modified": [{"uuid": "05f46699-5a78-4b28-9747-3f755f47eb2b", "delete_exceptions": [1255676500000, 1256803300000]}]
```

Das Feld „delete_exceptions“ soll gesetzt und geschickt werden ausschließlich wenn **neuen** Ausnahmen erzeugt werden sollen. Bereits bestehende Ausnahmen sollen nicht mitgeschickt werden. **TODO** Anderes Verhalten implementieren und beschreiben

„**change_exceptions**“ enthält ein JSONArray, welches geänderte Einzel-Termine(Ausnahmen) der Serie enthält. Jeder Termin in dieser Liste ist selbst ein JSONObject und enthält alle Felder wie ein normaler Termin und eine eigene UUID. Ähnlich wie bei den gelöschten Terminen kann man auch hier die Ausnahmen bei der Generierung oder Änderung einer Serie mitschicken, z.B.:

```
"created": [{"interval": 1, "color_label": 0, "created_by": "7", "end_date": "1255684700000", "folder_id": "41", "full_time": false, "id": "2465", "modified_by": "7", "note": "Do not delete!", "objectType": "calendar", "participants": [{"id": 7, "type": 1}], "private_flag": false, "recurrence_type": 1, "shown_as": 1, "start_date": "1255683700000", "title": "Serie", "users": [{"confirmation": 1, "confirmmessage": "", "display_name": "", "id": 7}], "uuid": "05f46699-5a78-4b28-9747-3f755f47eb2b", "change_exceptions": [{"end_date": "1255689700000", "recurrence_position": 3, "start_date": "1255687700000", "title": "Exception1", "uuid": "05f46699-5a78-4b28-9747-3f755f47eb2c"}, {"end_date": "1255699700000", "recurrence_position": 4, "start_date": "1255697700000", "title": "Exception2", "uuid": "05f46699-5a78-4b28-9747-3f755f47eb2d"}]}
```

oder:

```
"modified": [{"uuid": "05f46699-5a78-4b28-9747-3f755f47eb2b", "change_exceptions": [{"end_date": "1255689700000", "note": "Exception to delete!", "recurrence_position": 3, "start_date": "1255687700000", "title": "Exception", "uuid": "05f46699-5a78-4b28-9747-3f755f47eb2c"}]}]
```

Änderungen an den Ausnahmen sollen als Änderung der Serie geschickt werden, z.B.:

```
"modified": [{"uuid": "05f46699-5a78-4b28-9747-3f755f47eb2b", "change_exceptions": [{"end_date": "1255689700000", "recurrence_position": 3, "start_date": "1255687700000", "title": "Exception modified", "uuid": "05f46699-5a78-4b28-9747-3f755f47eb2c"}]}]
```

Wie in den Beispielen zu sehen ist, ist die Ausnahme selbst ein JSONObject mit eigener UUID:

```
{"end_date": "1255689700000", "recurrence_position": 3, "start_date": "1255687700000", "title": "Exception1", "uuid": "05f46699-5a78-4b28-9747-3f755f47eb2c"}
```

Ausnahmen, deren UUIDs dem Server noch nicht bekannt sind, müssen das Feld „recurrence_position“ enthalten um den genauen Termin der geändert werden soll zu bestimmen. Das Feld „recurrence_position“ ist in der Tabelle „Detailed task and appointment data“ genauer beschrieben.

Der Client soll in dem Feld „change_exceptions“ nur die Ausnahmen schicken die sich seit dem letzten Sync geändert haben. **TODO** Geändertes Verhalten implementieren und beschreiben.

Bitte beachten:

- der Client kann Ausnahmen erzeugen und ändern, aber nicht löschen.
- Ausnahmen werden vom OX-Server gelöscht, wenn eine Änderung durchgeführt wird, die den Zeitraum (Dauer der gesamten Serie oder Dauer jedes einzelnen Termins) der Serie betrifft, z.B:

syncUpdate 1 (Create Series):

```
{
  "sessionid":!!sessionid!!,
  "syncid": !!sync1!!,
  "folderid":"05f46699-5a78-4b28-9747-3f755f47eb2a",
  "created":{
    "interval":1,
    "color_label": 0,
    "created_by": "7",
    "end_date": 1255684700000,
    "folder_id": "41",
    "full_time": false,
    "id": "2465",
    "modified_by": "7",
    "note": "Do not delete!",
    "objectType": "calendar",
    "participants": [
      {
        "id": 7,
        "type": 1
      }
    ],
    "private_flag": false,
    "recurrence_type":1,
    "shown_as": 1,
    "start_date": 1255683700000,
    "title": "Serie",
    "users": [
      {
        "confirmation": 1,
        "confirmmessage": "",
        "display_name": "",
        "id": 7
      }
    ],
    "uuid": "!!serie_uuid!!"
  }
}
```

syncUpdate 2 (Create Exception):

```
{
  "sessionid":!!sessionid!!,
  "syncid": !!sync3!!,
  "folderid":"05f46699-5a78-4b28-9747-3f755f47eb2a",
  "modified":{
    "uuid": !!serie_uuid!!,
    "change_exceptions": [
      {
        "end_date": 1255689700000,
        "note": "Exception to delete!",
        "recurrence_position": 3,
        "start_date": 1255687700000,
        "title": "Exception",
        "uuid": "!!ex_uuid!!"
      }
    ]
  }
}
```

syncUpdate 3 (Modify Series – change start and end date):

```
{
  "sessionid":!!sessionid!!,
  "syncid": !!sync6!!,
  "folderid":"05f46699-5a78-4b28-9747-3f755f47eb2a",
  "modified":{
    "end_date": 1255696000000,
    "start_date": 1255690000000,
    "uuid": !!serie_uuid!!
  }
}
```

syncUpdate 4 (Get the current status):

```
{
  "sessionid":!!sessionid!!,
  "syncid":!!sync7!!,
  "folderid":"05f46699-5a78-4b28-9747-3f755f47eb2a"
}
```

Der Response beim syncUpdate 4 ist:

```
{
  "data": {
    "modified":!!serie!!,
    "deleted":!!ex_uuid!!,
    "syncid": !!sync8!!,
    "status": 1
  }
}
```

d.h. die Serie ist geändert worden und damit ist die Ausnahme gelöscht worden.

ContactContentType

Identifikator

contacts

ContentType Daten

Im Falle ContactContentType bestehen die Daten aus **Common object data** und Detailed contact data.

Detailed contact data

ID	Name	Type	Value
500	display_name	String	
501	first_name	String	First name.
502	last_name	String	Last name.
503	second_name	String	

504	suffix	String	
505	title	String	
506	street_home	String	
507	postal_code_home	String	
508	city_home	String	
509	state_home	String	
510	country_home	String	
511	birthday	Date	
512	marital_status	String	
513	Number_of_ children	String	
514	profession	String	
515	nickname	String	
516	spouse_name	String	
517	anniversary	Date	
518	note	String	
519	department	String	
520	position	String	
521	employee_type	String	
522	room_number	String	
523	street_business	String	
525	postal_code_business	String	
526	city_business	String	
527	state_business	String	
528	country_business	String	
529	number_of_employees	String	
530	sales_volume	String	
531	tax_id	String	
532	commercial_register	String	
533	branches	String	

534	business_category	String	
535	info	String	
536	manager_name	String	
537	assistant_name	String	
538	street_other	String	
539	city_other	String	
540	postal_code_other	String	
541	country_other	String	
542	telephone_business1	String	
543	telephone_business2	String	
544	fax_business	String	
545	telephone_callback	String	
546	telephone_car	String	
547	telephone_company	String	
548	telephone_home1	String	
549	telephone_home2	String	
550	fax_home	String	
551	cellular_telephone1	String	
552	cellular_telephone2	String	
553	telephone_other	String	
554	fax_other	String	
555	email1	String	
556	email2	String	
557	email3	String	
558	url	String	
559	telephone_isdn	String	
560	telephone_pager	String	
561	telephone_primary	String	
562	telephone_radio	String	

563	telephone_telex	String	
564	telephone_ttytd	String	
565	instant_messenger1	String	
566	instant_messenger2	String	
567	telephone_ip	String	
568	telephone_assistant	String	
569	company	String	
570	image1	String	
571	userfield01	String	
572	userfield02	String	
573	userfield03	String	
574	userfield04	String	
575	userfield05	String	
576	userfield06	String	
577	userfield07	String	
578	userfield08	String	
579	userfield09	String	
580	userfield10	String	
581	userfield11	String	
582	userfield12	String	
583	userfield13	String	
584	userfield14	String	
585	userfield15	String	
586	userfield16	String	
587	userfield17	String	
588	userfield18	String	
589	userfield19	String	
590	userfield20	String	
592	distribution_list	Array	If this contact is a distribution list, then

			this field is an array of objects. Each object describes a member of the list as defined in Distribution list member
594	number_of_distribution_list	Number	
596	number_of_images	Number	
598	state_other	String	
599	file_as	String	
602	mark_as_distributionlist	Boolean	
605	default_address	Number	
524	internal_userid	Number	

Distribution list member

Name	Type	Value
id	String	User ID of the member if the member is an existing contact.
first_name	String	First name
last_name	String	Last name
mail	String	Email address
mail_field	Number	Which email field of an existing contact (if any) is used for the mail field. 0 independent contact 1 default email field (email1) 2 second email field (email2) 3 third email field (email3)

MailContentType

Identifikator

mail

ContentType Daten

Im Falle des MailContentType weichen die vom USM verwendeten Daten von den Daten der OX-JSON-API ab. Folgende Felder aus der Standard-OX-JSON-API werden verwendet:

Detailed mail data

ID	Name	Type	Value																
102	color_label	Number	Color number used by Outlook to label the object. The assignment of colors to numbers is arbitrary and specified by the client. The numbers are integer numbers between 1 and 10 (inclusive).																
600	id	String	Object ID																
601	folder_id	String	Object ID of the parent folder																
611	flags	Number	<div>Various system flags. A sum of zero or more of following values:</div> <table><tr><td>1</td><td>answered</td></tr><tr><td>2</td><td>deleted</td></tr><tr><td>4</td><td>draft</td></tr><tr><td>8</td><td>flagged</td></tr><tr><td>16</td><td>recent</td></tr><tr><td>32</td><td>seen</td></tr><tr><td>64</td><td>user</td></tr><tr><td>256</td><td>forwarded</td></tr></table> <div>See javax.mail.Flags.Flag for details.</div>	1	answered	2	deleted	4	draft	8	flagged	16	recent	32	seen	64	user	256	forwarded
1	answered																		
2	deleted																		
4	draft																		
8	flagged																		
16	recent																		
32	seen																		
64	user																		
256	forwarded																		

Zusätzlich werden eigene Felder bereitgestellt, die die MIME-Struktur der Email repräsentieren. Außerdem wird ein besonderes Feld ("meeting_request"/"meeting_update") gefüllt, falls die Email als Einladung zu einem Termin erkannt wurde, oder „task_request“/“task_update“ falls die Email als Einladung zu einer Aufgabe erkannt wurde. In diesem Feld werden alle Daten eines Termins / Aufgabe (inkl. UUID und UID) als Felder übergeben:

Name	Type	Value
body	JSONObject oder JSONArray	Either a simple mail body (JSONObject), an array of multipart mail parts (JSONArray) or itself a mail object (JSONObject)
headers	JSONObject	The mail's headers. For each header a corresponding field exists. In case of multi-value header a JSON array of strings is returned
received_date	Number	The mail system's internal time stamp of mail delivery as a UTC long value
User	JSONArray	An array of mail's user flags as strings; e.g. ["\$Custom5", "\$Work"]

meeting_request (meeting_update)	JSONObject	An appointment object with all fields set for an appointment to which this email is detected as an invitation/update
task_request (task_update)	JSONObject	A task object with all fields set for a task to which this email is detected as an invitation/update

Für das Feld "body" gibt es zwei Möglichkeiten:

Simple Mail Body

Name	Type	Value
id	String	The part's sequence identifier in parental mail
data	String	The base64 encoded data of corresponding mail part

Multipart Mail Body

Name	Type	Value
body	JSONObject oder JSONArray	Contains current mail part's body
headers	JSONObject	The part's headers. For each header a corresponding field exists. In case of multi-value header a JSON array of strings is returned

Weiterführende Informationen und Beispiele sind auch hier zu finden:

<https://dev-wiki.open-xchange.com/cgi-bin/twiki/view/Main/StructuredJSONMailObject>

GroupContentType

Identifikator

groups

ContentType Daten

Im Falle GroupContentType bestehen die Daten nur aus *Group data*.

Group data

	Name	Type	Value
	id	String	ID
	display_name	String	Display name
	name	String	Name with character restrictions
	members	Array	Array of group member id's

ResourcesContentType

Identifikator

resources

ContentType Daten

Im Falle ResourcesContentType bestehen die Daten nur aus *Resource Response*.

Resource response

	Name	Type	Value
	id	Number	ID
	display_name	String	Display name

Anhang C: OX Content Types

Dieser Anhang enthält die OX Data Typen die im USM nicht abgebildet sind.

InfoitemContentType

Identifikator

infoitems

ContentType Daten

Im Falle InfoitemContentType bestehen die Daten aus **Common object data** und *Detailed infoitem data*.

Detailed infoitem data

	ID	Name	Type	Value
700		title	String	Title
701		url	String	Link/URL
702		filename	String	Displayed filename of the document.

703	file_mimetype	String	MIME type of the document. The client converts known types to more readable names before displaying them.
704	file_size	Number	Size of the document in bytes.
705	version	Number	Version number of the document. New documents start at 1. Every update increments the version by 1.
706	description	String	Description
707	locked_until	Time	The time until which this item will presumably be locked. Only set if the document is currently locked, 0 otherwise.
708	file_md5sum	String	MD5Sum of the document. Not yet implemented, so this is currently always empty.
709	version_comment	String	A version comment is used to file a changelog for the file.
710	current_version	Boolean	"true" if this version is the current version "false" otherwise. . Note: This is not writeable
711	number_of_versions	Number	The number of all versions of the infoitem. Note: This is not writeable.

144

144