



Beschreibung des Benno MailArchiv Application Programming Interface (API)

- Web Service API -

für

Benno MailArchiv Version 2.x

Dokumentversion:	0.9
Datum der letzten Änderung:	02.11.2012
Letzte Änderung wurde durchgeführt von:	Ansgar Licher
Datum der Erstellung dieser Dokumentation:2	22.06.2012
Die Ursprungsversion dieser Dokumentation wurde erstellt durch:	Dipl.-Ing. Martin Werthmüller LWsystems GmbH & Co. KG

Beschreibung des Benno MailArchiv Application Programming Interface (API) / Web Service API

Gültigkeit:

Die in dieser Installationsanleitung dargelegten Informationen und Vorgänge beziehen sich auf den Benno MailArchiv Release-Stand 2.x.

Copyright © 2009-2012 LWsystems GmbH & Co. KG

Dieses Werk ist geistiges Eigentum der LWsystems GmbH & Co. KG. Das Werk darf als Ganzes oder auszugsweise kopiert werden, vorausgesetzt, dass dieser Copyright-Vermerk in jeder Kopie enthalten ist.

Die in diesem Buch enthaltenen Informationen wurden mit größter Sorgfalt zusammengestellt. Fehlerhafte Angaben können jedoch nicht vollkommen ausgeschlossen werden. Die LWsystems GmbH & Co. KG sowie die Autoren haften nicht für eventuelle Fehler und deren Folgen.

Die in dieser Anleitung verwendeten Soft- und Hardwarebezeichnungen sind in der Regel auch eingetragene Warenzeichen; sie werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Die LWsystems GmbH & Co. KG richtet sich im Wesentlichen nach den Schreibweisen der Hersteller. Die Wiedergabe von Waren- und Handelsnamen usw. in diesem Buch (auch ohne besondere Kennzeichnung) berechtigt nicht zu der Annahme, dass solche Namen (im Sinne der Warenzeichen und Markenschutz-Gesetzgebung) als frei zu betrachten sind.

Inhaltsverzeichnis

0 Allgemeines.....	4
I Generelle Informationen.....	5
I.1 Sicherheitsaspekte.....	6
I.1.1 Authentisierung über das Shared Secret.....	6
I.1.2 Reichweite von Suchanfragen.....	6
I.1.3 Zugriff auf E-Mails über deren eindeutige ID im Archiv.....	7
II API Calls.....	8
II.1 API-Call: Durchführung einer Suchanfrage.....	8
II.1.1 Syntaxbeispiel für Suchanfragen.....	8
II.1.2 Parameter einer Suchanfrage.....	9
II.1.3 Rückgabewerte einer Suchanfrage.....	10
II.1.4 Beispiele für Suchanfragen.....	10
II.1.4.1 Allgemeines.....	10
II.1.4.2 BASIC AUTH durch Übergabe von Benutzer und Shared Secret über den URL.....	11
II.1.4.3 Minimale Suchanfrage.....	11
II.1.4.4 Suchanfrage mit zwei Suchkriterien, per UND verknüpft.....	11
II.1.4.5 Suchanfrage mit Filterung.....	11
II.1.4.6 Suchanfrage mit zwei Suchkriterien, per ODER verknüpft.....	11
II.1.5 Suchkriterien.....	11
II.2 API-Call: Zugriff auf eine E-Mail im Archiv.....	12
II.2.1 Syntaxbeispiel für den Abruf einer E-Mail aus dem Archiv.....	12
II.2.2 Parameter für den Abruf einer E-Mail aus dem Archiv.....	12
II.2.3 Rückgabewerte eines Abrufs einer E-Mail aus dem Archiv.....	13
II.2.4 Beispiele für den Abruf von Mails aus dem Archiv.....	13
III Anlagen.....	14
III.1 Suchkriterien in Suchanfragen.....	14
III.2 Rückgabe der Ergebnisse einer Suchanfrage.....	14
III.3 Abruf und Rückgabe einer E-Mail.....	17

0 Allgemeines

Diese Dokumentation der Benno MailArchiv API (API – Application Programming Interface) beschreibt die von Benno MailArchiv zur Verfügung gestellte Web Service Programmierschnittstelle. Bei der API handelt es sich um ein sog. „**REST-API**“ bzw. eine API auf der Basis der „RESTful“ Technologie.

Mittels dieser API können die Anwenderfunktionen von Benno MailArchiv, die über die WebApp zur Verfügung stehen, unabhängig von dieser aus Drittapplikation heraus ausgeführt werden. Die API ist eine **Web-Service API**, d.h., dass die über die API bereitgestellten Funktionalitäten als **Web-Service für Drittanwendungen** bzw. in Form von **SOA** zur Verfügung stehen.

Folgende Funktionen stehen über die API zur Verfügung:

- E-Mails suchen (eine Suchanfrage an Benno MailArchiv übergeben)
- Trefferliste (Trefferliste auf eine Suchanfrage hin als Rückgabe erhalten)
- Mail anzeigen (eine konkrete E-Mail aus dem Archiv auslesen)

Weiterführende und allgemeine Informationen zu den Themen **REST** und **SOA** finden Sie im Internet, bspw. unter folgenden URLs:

http://de.wikipedia.org/wiki/Representational_State_Transfer

http://en.wikipedia.org/wiki/Representational_state_transfer

http://de.wikipedia.org/wiki/Serviceorientierte_Architektur

Diese API-Dokumentation ist ausschließlich gültig für die Softwareversion Benno MailArchiv 2.x.

Bei Fragen wenden Sie sich an:

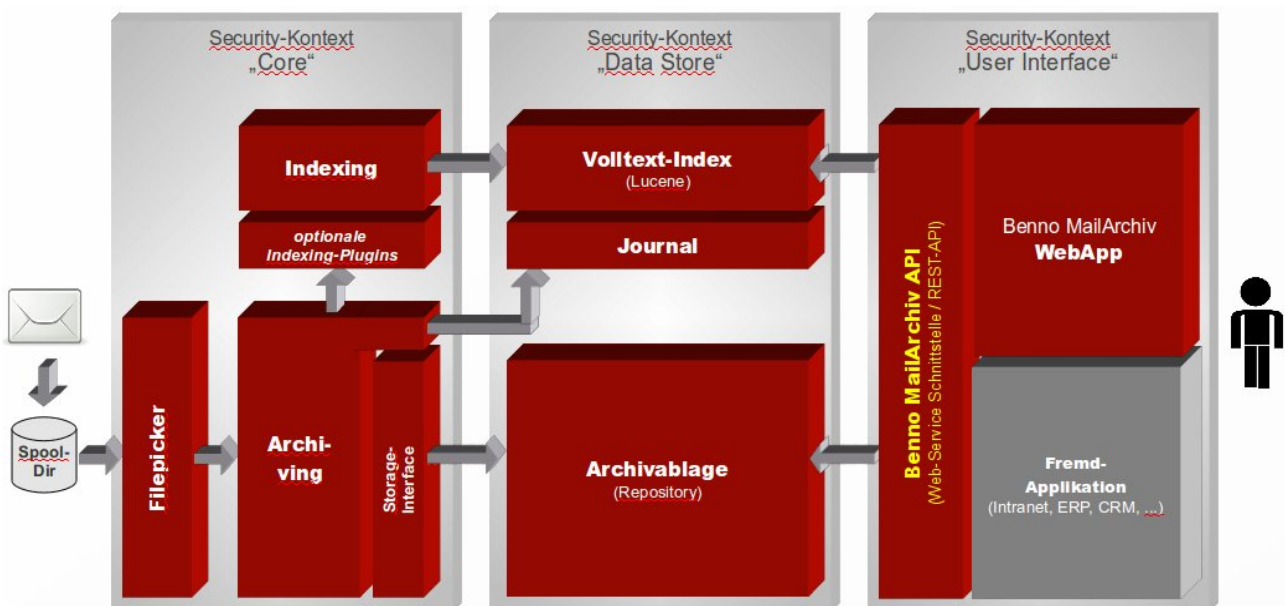
Kontaktdaten: Telefon: +49 (0) 5403 / 5556
 E-Mail: support@benno-mailarchiv.de

I Generelle Informationen

Die Benno MailArchiv API weist folgende generelle Eigenschaften auf:

- Die Rückgabewerte der Funktionsaufrufe werden im JSON-Format übergeben (bspw. wird die Liste der auf eine Suchanfrage hin gefundenen E-Mails als Liste von JSON-Objekten zurück gegeben).
- Die Rückgabewerte sind UTF-8 codiert.
- Die Rückgabe einer aus dem Archiv gelesenen bzw. angeforderten E-Mail kann sowohl im JSON Format (also als Array von JSON Objekten) oder im „raw“-Format (also RFC 822, Plaintext) erfolgen.
- Anfragen über die API an den Benno MailArchiv Server sind aus Sicherheitsgründen nur möglich, wenn sich das aufrufende Programm gegenüber dem Server authentisiert. Die Authentisierung erfolgt mittels HTTP Basic Authentication.
Details siehe bspw.: <http://de.wikipedia.org/wiki/HTTP-Authentifizierung>
und weiter: <http://tools.ietf.org/html/rfc2617>
- Alle API-Calls gegenüber dem Benno MailArchiv Server sollten als HTTP POST-Requests ausgeführt werden.
- Jeglicher Zugriff aus der Benutzerperspektive auf Benno MailArchiv erfolgt ausschließlich über die API. D.h.: Außerhalb der API sind keine Zugriffsmöglichkeiten auf Benno MailArchiv implementiert. Auch die Benno MailArchiv WebApp, also die reguläre GUI für den Anwender, greift ausschließlich über diese API auf Benno MailArchiv zu.

Nachstehende Grafik verdeutlicht die Architektur von Benno MailArchiv und die Einbettung der API in das System:



Wie das obige Schaubild aufzeigt, ist Benno MailArchiv in einer Multi-Tier-Architektur implementiert. Das Benno MailArchiv Backend (bzw. der „Core“-Kontext) ist logisch vom Frontend („User Interface“-Kontext bzw. der API) getrennt. Diese Architektur erlaubt es prinzipiell, die beiden Kontexte (bzw. Backend und Frontend) netzwerktransparent auf unterschiedlichen Servern zu betreiben.

Da die beiden Kontexte theoretisch nicht auf dem gleichen Server laufen müssen, ist ein Zugriff auf das Backend durch ein zu vergebendes Shared Secret abgesichert. Ein Benno MailArchiv Server, auf dem das Frontend läuft, hat demgemäß nur dann Zugriff auf den Backend Server, wenn das Shared Secret auf beiden Seiten identisch ist.

I.1 Sicherheitsaspekte

Bei der Verwendung der Benno MailArchiv API sind folgende Sicherheitsaspekte unbedingt zu bedenken.

I.1.1 Authentisierung über das Shared Secret

Das Shared Secret ist der „Schlüssel“, um über das Frontend bzw. die API Zugriff auf das Backend des Benno MailArchiv Servers zu erhalten. Fällt das Shared Secret in falsche Hände, ist das Backend (und damit die Summe der archivierten E-Mails) nicht mehr wirksam vor Fremdzugriffen geschützt. Das Shared Secret selbstverständlich kann jederzeit einfach geändert werden.

Die Authentisierung mit dem Shared Secret erfolgt (wie oben erwähnt) als HTTP Basic Auth.

Wir warnen eindringlich davor, das Shared Secret in browser-basierten Web-Anwendungen im URL zu übergeben! Das Shared Secret ist hier unverschlüsselt sichtbar und kann so leicht in falsche Hände gelangen!

Wir empfehlen, die Authentisierung gegen das REST-API so zu gestalten, dass der Benutzer keinerlei Möglichkeiten hat, diese (bspw. über den URL usw.) einzusehen oder zu manipulieren. Die Implementierung sollte hier anhand eines Proxy oder eines ähnlichen Entwurfsmusters erfolgen. Konkret kann dieses z.B. sehr einfach durch einen vorgeschalteten HTTP-Server mit einem CGI-Script durchgeführt werden.

***Hinweis:** Die HTTP Basic Auth Authentifizierung basiert auf den beiden Faktoren „Username“ und „Password“. Als Username ist grundsätzlich „benno2“ zu verwenden. Das Passwort ist das auf dem Server hinterlegte Shared Secret.*

I.1.2 Reichweite von Suchanfragen

Die API greift (im Gegensatz zur Benno MailArchiv WebApp!) uneingeschränkt auf alle E-Mails des Archivs zu. D.h., dass jede Suchanfrage Treffer unabhängig von etwaigen Benutzerberechtigungen, die in der WebApp eingerichtet sind, zurückliefert!

Mittels entsprechender Filterparameter **kann** eine Suchanfrage auf bestimmte E-Mails eingeschränkt werden (bspw. E-Mails eines bestimmten Benutzers). Die Filterparameter sind (soweit eine solche Filterung erfolgen soll) beim API Aufruf anzugeben.

Die API selbst greift **nicht!** auf die Benno MailArchiv Benutzerverwaltung der WebApp zurück! Die Durchsetzung der Filterung bzw. die Durchsetzung von Zugriffsbeschränkungen ist durch das aufrufende Programm, das diese API verwendet, selber zu implementieren!

Weiter unten In dieser Dokumentation wird erläutert, wie eine Einschränkung der Suche auf bestimmte Mails (bspw. die eines bestimmten Benutzers) durch die Verwendung der Filterung vorgenommen werden kann.

I.1.3 Zugriff auf E-Mails über deren eindeutige ID im Archiv

Alle archivierten E-Mails haben eine eindeutige ID, über die der direkte Zugriff auf die jeweilige E-Mail möglich ist. Ähnlich wie das Shared Secret stellt diese eindeutige ID einen sicherheitsrelevanten Aspekt dar.

Da eine ID eindeutig eine bestimmte E-Mail im Archiv adressiert, sollte diese ID ebenso wenig wie das Shared Secret direkt an den Benutzer weiter gegeben werden! Die Kenntnis der ID ermöglicht den Zugriff auf die zugehörige E-Mail im Archiv.

Hinweis: In der Benno MailArchiv WebApp werden die tatsächlichen IDs der E-Mails der Trefferliste vor dem Anwender verborgen, in dem sie in der jeweiligen Session in einer Schattentabelle auf temporäre IDs umgemappt werden. Sieht ein Anwender die IDs der E-Mails einer Trefferliste, so sind diese wertlos, da sie nicht den Original-IDs der archivierten Mails entsprechen. Ein direkter Zugriffsversuch auf eine Mail mit einer solchen temporären ID führt daher ins Leere.

I.1.4 SSL-Verschlüsselung

Benno MailArchiv unterstützt selbstverständlich HTTP sowie HTTPS, die SSL-verschlüsselte Variante. In dieser Dokumentation sind alle URLs als HTTP URLs dargestellt. Sie können selbstverständlich ohne jegliche Änderungen mit HTTPS verwendet werden.

Abschließender Hinweis:

Es ist dem Softwareentwickler bzw. Programmierer, der diese API verwendet, selbstverständlich frei gestellt, diese sicherheitsrelevanten Hinweise zu ignorieren, wir raten jedoch dazu, sie insbes. dann zu beachten, wenn die API zur Integration der Benno MailArchiv-Funktionen in Web-Anwendungen erfolgen soll, um eine nachhaltig sichere und unkompromittierbare Lösung zu entwickeln.

II API Calls

II.1 API-Call: Durchführung einer Suchanfrage

Suchanfragen dienen zur Suche nach archivierten E-Mails. Da alle E-Mails incl. ihrer Attachements volltext-indexiert vorliegen, können E-Mails anhand jeden beliebigen Inhalts bzw. anhand jedes beliebigen Merkmals (und selbstverständlich auch beliebiger Kombinationen von Merkmalen) gefunden werden.

Suchanfragen werden mittels eines HTTP URL aufgerufen. Die Syntax ist dabei grundsätzlich:

```
http://<bennohost>:8080/search/{suchanfrage}
```

Der dem Keyword „search“ folgende Slash (/) ist dabei obligatorisch. Die Suchanfrage selbst setzt sich aus mehreren Parametern zusammen. Diese werden in der üblichen Form in dem URL codiert:

```
?option1=wert1&option2=wert2&option3=wert3
```

Die vollständige Suchanfrage hat damit folgendes Format:

```
http://<bennohost>:8080/search/?option1=wert1&option2=wert2...
```

II.1.1 Syntaxbeispiel für Suchanfragen

***Hinweis:** Die URLs werden aus drucktechnischen Gründen teilweise mehrzeilig dargestellt. Sie sind jedoch ohne Leerzeichen usw. in einer Zeile anzugeben!*

Hier zunächst die syntaktische Darstellung von Suchanfragen. Weiter unten sind konkrete Beispiele für Suchanfragen abgebildet.

Die Query Syntax

Suchanfragen werden in Benno MailArchiv grundsätzlich in der Query-Syntax von Apache Lucene verfasst. Die vollständige Syntax von Lucene kann bspw. in den nachstehenden Quellen nachgelesen werden:

http://lucene.apache.org/core/old_versioned_docs/versions/2_9_1/queryparsersyntax.html

<http://www.lucenetutorial.com/lucene-query-syntax.html>

Minimale Suchanfrage

```
http://<bennohost>:8080/search/  
?archive=<container-identifier>&query=<suchanfrage>
```


Suchanfrage mit Filterung

```
http://<bennohost>:8080/search/  
?archive=<container-identifizier>&filterQuery=<filter>&query=<suchanfrage>
```

Die Verwendung von Filtern in Suchanfragen kann zu effizienterem Verarbeiten der Abfragen führen. Die Verarbeitungsreihenfolge in Lucene ist dabei:

- zunächst Ausführung der „query“,
- danach Filterung der Ergebnisliste mit der „filterQuery“.

***Hinweis:** Alle (auch komplexe) Suchanfragen können als „query“ gestaltet werden. Die Filterung kann bspw. ergänzend verwendet werden, um die Menge der Suchergebnisse auf E-Mails mit bestimmten Merkmalen einzugrenzen. Bspw. werden alle Suchanfragen der Benno MailArchiv WebApp über „query“ abgebildet, während die Benutzerberechtigungen durch Filter („filterQuery“) abgebildet werden.*

II.1.2 Parameter einer Suchanfrage

Pflichtparameter:

Eine Suchanfrage setzt sich aus mindestens folgenden Parametern zusammen:

- **archive:** eindeutiger Identifizier des Containers¹, in dem die Suche erfolgen soll
- **query:** Die eigentliche Suchanfrage

Suchanfragen mit Filterung haben darüber hinaus den zusätzlich Parameter

- **filterQuery:** Filterkriterien, die nach der Suche auf das Suchergebnis angewendet werden

Optionale Parameter:

Suchanfragen können neben den o.g. Pflichtparametern zusätzlich folgende optionale Parameter enthalten:

- **start:** erstes auszugebendes Element des Ergebnisses (Standard: 1) (eins-basiert)
- **limit:** Anzahl der auszugebenden Elemente des Ergebnisses (Standard: 20)
- **sort:** Attribut, nach dem die Trefferliste sortiert werden soll (Standard: „sent“ (Sendedatum))

¹ Container: Benno MailArchiv ist vollständig Multi-Mandantenfähig. Mehrere Mandanten können parallel in einer einzigen Benno MailArchiv Instanz abgebildet werden. Jeder Mandant wird dabei in einem separaten Container abgebildet. Jeder Container erhält einen eindeutigen Namen, über den er eindeutig identifiziert werden kann. Container werden ihrerseits in Boxen unterteilt. Die E-Mails selbst werden dabei in den Boxen abgelegt. Boxen beinhalten die E-Mails eines Zeitraums (bspw. Woche, Monat, Quartal, Jahr usw.).
Eine Benno MailArchiv Standard-Installation (also eine Installation „out of the box“) umfasst einen Mandanten (und damit einen Container), in den alle E-Mails (in Boxen unterteilt) archiviert werden. Der Name des Standard-Containers einer solchen Installation ist vorgegeben und heißt „**BennoContainer**“. Der Containername ist in der Benno MailArchiv Konfiguration änderbar.

- **up**: Sortierung aufsteigend (Standard: false)
- **full**: Ausgabe aller Attribute der Maildokumente (Standard: false)

II.1.3 Rückgabewerte einer Suchanfrage

Resultatwerte (HTTP-Fehlercodes der Suchoperation):

- 1xx - Information
- 2xx - OK ...
- 3xx - Aufforderungen
- 4xx - Temporäre Fehler
- 5xx - Dauerhafte Fehler

Resultatdaten (Suchergebnisse bzw. Trefferlisten):

- **limit**: wie aus den Parametern, oder die Defaultwerte
- **found**: Anzahl der gefundenen E-Mails
- **first**: Stelle im Suchergebnis der ersten ausgegebenen (Zeile(eins-basiert), 0 falls keine Zeile ausgegeben wurde)
- **last**: Stelle im Suchergebnis der letzten ausgegebenen (Zeile(eins-basiert), 0 falls keine Zeile ausgegeben wurde)
- **count**: Anzahl der Zeilen in dieser Rückgabe
- **results**: Trefferliste; Array mit JSON Objekten mit dem folgenden Aufbau:
 - **from**: Absenderadresse
 - **to**: Zieladressen
 - **multipart**: 1 wenn Mehrteilig, 0 wenn nicht
 - **subject**: Betreff
 - **summary**: Zusammenfassung
 - **sent**: das Absendedatum
 - **id**: eindeutige ID der archivierten E-Mail (notwendig, um diese aus dem Archiv abrufen zu können)

II.1.4 Beispiele für Suchanfragen

II.1.4.1 Allgemeines

Die nachfolgenden Beispiele für Suchanfragen beginnen alle mit dem folgenden HTTP URL. Dieser URL wird in den Beispielen nicht weiter extra aufgeführt.

```
http://<bennohost>:8080/search/
```

Weiterhin ist zu beachten, dass die Suchanfragen grundsätzlich einzeilig sind, auch wenn sie nachstehend oft mehrzeilig dargestellt werden!

II.1.4.2 BASIC AUTH durch Übergabe von Benutzer und Shared Secret über den URL

Wie bereits eingangs erwähnt, raten wir **ausdrücklich** dazu, diese Form der Autorisierung **nicht** in browser-basierten Web-Anwendungen einzusetzen, da das Shared Secret im Klartext in der URL enthalten ist bzw. im Browser nachgelesen werden kann!

```
http://benno2:<shared-secret>@localhost:8080/search/...
```

II.1.4.3 Minimale Suchanfrage

```
?archive=BennoContainer&query=(Sender:*hansen-und-meyer*)
```

In dieser Suchanfrage werden alle E-Mails im Standard-Container² gesucht, die von Absenderadressen stammen, in denen das Literal „hansen-und-meyer“ vorkommt.

II.1.4.4 Suchanfrage mit zwei Suchkriterien, per UND verknüpft

```
?archive=BennoContainer&query=((Sender:*hansen-und-meyer*)) AND  
(Recipient:vertrieb@*)
```

In dieser Suchanfrage werden alle E-Mails im Standard-Container gesucht, die von Absenderadressen stammen, in denen das Literal „hansen-und-meyer“ vorkommt UND deren Empfängeradressen das Literal „vertrieb@“ enthalten.

II.1.4.5 Suchanfrage mit Filterung

```
?archive=BennoContainer&filterQuery=(Recipient:vertrieb@*)  
&query=(Sender:*hansen-und-meyer*)
```

Bei dieser Suchanfrage werden ebenfalls alle Mails gesucht, deren Absenderadresse das Literal „hansen-und-meyer“ enthält. Anschließend werden die Ergebnisse gefiltert. Es werden alle E-Mails ausgefiltert, deren Empfängeradressen nicht „vertrieb@“ enthalten.

Wie gesehen, führen beide Wege zum gleichen Ergebnis.

² Standard-Container: In einer Benno MailArchiv Standard-Installation („out of the box“) ist automatisch ein Archiv mit einem Container (= ein Mandant) enthalten und vorkonfiguriert. Dies ist der Standard-Containers. Der Name des Standard-Containers ist vorgegeben und heißt „BennoContainer“ (siehe auch Fußnote 1).

II.1.4.6 Suchanfrage mit zwei Suchkriterien, per ODER verknüpft

```
?archive=BennoContainer&query=((Sender:*hansen-und-meyer*))  
(Recipient:vertrieb@*)
```

In dieser Suchanfrage werden alle E-Mails im Standard-Container gesucht, die von Absenderadressen stammen, in denen das Literal „hansen-und-meyer“ vorkommt ODER deren Empfängeradressen das Literal „vertrieb@“ enthalten.

II.1.5 Suchkriterien

Neben den Suchkriterien für `Sender` bzw. `Recipient` stehen weitere Suchkriterien zur Verfügung. Eine Liste aller Suchkriterien befindet sich im Anhang dieser Dokumentation (siehe Abschnitt „Suchkriterien in Suchabfragen“).

***Hinweis:** Alle Suchkriterien können in Suchanfragen in beliebiger Form (UND, ODER) und in beliebiger Reihenfolge miteinander kombiniert werden.*

II.2 API-Call: Zugriff auf eine E-Mail im Archiv

Der API-Call zum Zugriff auf eine archivierte E-Mail dient dazu, eine E-Mail (incl. etwaiger Attachments) aus dem Archiv zu holen. Da jede E-Mail im Archiv eindeutig identifiziert ist, erfolgt der Zugriff über diesen eindeutigen Identifier.

II.2.1 Syntaxbeispiel für den Abruf einer E-Mail aus dem Archiv

***Hinweis:** Die URLs werden aus drucktechnischen Gründen teilweise mehrzeilig dargestellt. Sie sind jedoch ohne Leerzeichen usw. in einer Zeile anzugeben!*

Hier zunächst die syntaktische Darstellung eines Abrufs. Weiter unten sind konkrete Beispiele für das Abrufen von E-Mails abgebildet.

Der Aufruf erfolgt analog der zu Suchanfragen als HTTP URL. Die Syntax ist dabei grundsätzlich:

```
http://<bennohost>:8080/mail/  
?archive=<container-identifier>&id=<id>
```

II.2.2 Parameter für den Abruf einer E-Mail aus dem Archiv

Pflichtparameter:

Der Abruf einer E-Mail setzt sich aus den beiden folgenden Parametern zusammen:

- **archive:** eindeutiger Identifier des Containers, aus dem die E-Mail gelesen werden soll
- **id:** Der eindeutige Identifier der E-Mail (ID) (incl. der Bezeichnung der Box im Container)

Optionale Parameter:

Die aus dem Archiv gelesene E-Mail wird standardmäßig als JSON Objekt an das aufrufende Programm zurückgegeben. Mittels der Option `format` kann die Rückgabe der E-Mail im „raw“ Format, also RFC 822 (Plaintext) erzwungen werden.

- **format:** „raw“, wenn die Mail im RFC-822 Format zurückgegeben werden soll.

II.2.3 Rückgabewerte eines Abrufs einer E-Mail aus dem Archiv

Resultatwerte (HTTP-Fehlercodes der Suchoperation):

- 1xx - Information
- 2xx - OK ...
- 3xx - Aufforderungen
- 4xx - Temporäre Fehler
- 5xx - Dauerhafte Fehler

Rückgabewerte:

Die E-Mail wird in Form eines Arrays mit JSON Objekten zurückgegeben:

- **header:** JSON Array mit Header-Zeilen
- **body:** Decodierter Body der E-Mail (wenn der Body bspw. base64 encodiert vorliegt)
- **attachments:** JSON Array mit Metadaten und Content der MIME-Parts

II.2.4 Beispiele für den Abruf von Mails aus dem Archiv

```
http://localhost:8080/mail/  
?archive=BennoContainer&id=2012:1C6626F53B0B...67ADA0 FB4E
```

Dieser Abruf liest die E-Mail mit der eindeutigen ID „1C6626F53B0B[...]67ADA0 FB4E“ aus dem Archiv und gibt sie an das aufrufende Programm zurück. Die E-Mail wird dabei aus der Box „2012“ des Containers „BennoContainer“ gelesen.

III Anlagen

III.1 Suchkriterien in Suchanfragen

E-Mails können anhand aller indextierten Inhalte gesucht werden. Die Suchkriterien können einzeln oder mit den logischen Operatoren „UND“ bzw. „ODER“ kombiniert werden. Details zur Syntax siehe oben und in der einschlägigen Dokumentation zu Apache Lucene im Internet (URLs siehe oben).

Folgende Suchkriterien stehen über die API für Suchanfragen zur Verfügung:

- **Date:** Datum der E-Mail
- **From:** Absenderadresse der E-Mail
- **Recipient:** Empfängeradressen (beinhaltet To, Cc und Bcc)
- **RecipientDomain:** Domäne des Empfängers
- **Sender:** Absenderadresse der E-Mail
- **SenderDomain:** Domäne des Absenders
- **Size:** Größe der E-Mail
- **Subject:** Betreffzeile der E-Mail
- **To:** To-Empfängeradressen
- **hasAttachment:** E-Mail mit Attachment
- **id:** Eindeutige ID der E-Mail im Archiv
- **Cc:** Cc-Empfängeradressen
- **Text:** Suche nach Wörtern im Text der E-Mail (incl. Text aus den Attachments)
- **HEADER-XYZ:** Suche nach E-Mails, bei denen der Mailheader „XYZ“ einen bestimmten Wert enthält

III.2 Rückgabe der Ergebnisse einer Suchanfrage

Die nachstehend aufgeführte Suchanfrage ergibt bspw. eine Ergebnismenge von 2 gefundenen E-Mails:

```
http://bennohost:8080/search/  
?archive=BennoContainer&query=(Sender:*hansen-und-meyer*)
```

Das Suchergebnis (2 Treffer) wird in Form von JSON Objekten zurückgegeben und hat bspw. folgenden Inhalt:

```
{
  "licenseValid":true,
  "limit":2,
  "results"
  {
    "id":"2012:748B034C4AF3004231D24B9DE2515822AB16485914AB3AE055C40DA336F4272F00",
    "Subject":"Re: Anfrage bezüglich Partnerschaft",
    "Date":"201206191122",
    "hasAttachment":"1",
    "To"
      "\"W. Robert\" <w.robert@robert-druck.de>"
    ],
    "Bcc"],
    "Cc"],
    "From":"info@hansen-und-meyer.de"
  },
  {
    "id":"2012:1C6626F5E02E1B3B0B74DE794BC67408A64484C8B8B98A83C130162B077AD6AB00",
    "Subject":"Re: Anfrage bezüglich Partnerschaft",
    "Date":"201206191122",
    "hasAttachment":"1",
    "To"
      "w.robert@robert-druck.de"
    ],
    "Bcc"],
    "Cc"],
    "From":"info@hansen-und-meyer.de"
  }
  ],
  "last":1,
  "count":2,
  "overall":25,
  "first":0,
  "found":14
}
```

Die Repräsentation dieses Suchergebnisses in einer Programmiersprache wie bspw. PHP oder Perl könnte bspw. so aussehen:

```
$VAR1 = {
  'first' => 0,
  'count' => 2,
  'last' => 1,
  'found' => 14,
  'overall' => 25,
  'licenseValid' => 1,
  'limit' => 2,
  'results' => [
    {
      'Cc' => [],
      'Subject' => "Re: Anfrage bez\u{x}{c3}\x{bc}glich Partnerschaft",
      'Date' => '201206191122',
      'Bcc' => [],
      'hasAttachment' => '1',
      'id' =>
'2012:748B034C4AF3004231D24B9DE2515822AB16485914AB3AE055C40DA336F4272F00',
      'To' => [
        'W. Robert' <w.robert@robert-druck.de>
      ],
      'From' => 'info@hansen-und-meyer.de'
    },
    {
      'Cc' => [],
      'Subject' => "Re: Anfrage bez\u{x}{c3}\x{bc}glich Partnerschaft",
      'Date' => '201206191122',
      'Bcc' => [],
      'hasAttachment' => '1',
      'id' =>
'2012:1C6626F5E02E1B3B0B74DE794BC67408A64484C8B8B98A83C130162B077AD6AB00',
      'To' => [
        'w.robert@robert-druck.de'
      ],
      'From' => 'info@hansen-und-meyer.de'
    }
  ]
}
```


III.3 Abruf und Rückgabe einer E-Mail

Nachstehend ein Abruf einer E-Mail aus dem Archiv:

```
http://bennohost:8080/mail/  
&archive=BennoContainer  
&id=2012:1C6626F5E02E1B3B0B74D...A83C130162B077AD6AB00
```

Das Ergebnis (die E-Mail) wird standardmäßig in Form von JSON Objekten zurückgegeben und hat bspw. folgenden Inhalt:

```
{  
  'body' => [  
    {  
      'content-type' => 'text/plain',  
      'data' => "Sehr geehrter Herr Robert,
```

wir danken f\u00fcr das Interesse an unseren Produkten, das Sie mit Ihrer E-Mail gezeigt haben.

Wir haben dieser E-mail ein Angebot hinzugef\u00fct und hoffen, dass wir baldig diesbez\u00f6glich von Ihnen h\u00f6ren werden.

Unsere Gesch\u00e4ftsbedingungen entnehmen Sie bitte den beigef\u00fcten PDF-Dokument.

Bei weiteren Fragen stehen wir Ihnen selbstverst\u00e4ndlich auch telefonisch zur Seite. Sie k\u00f6nnen uns unter der Rufnummer 0541/123456-1, von Mo. bis Fr. zwischen 09:00 und 17:00 Uhr, erreichen.

Mit freundlichen Gr\u00fc\u00dfen,

Papierfabrik Hansen & Meyer GmbH

i. A.

Sabine Petersen

Papierfabrik Hansen & Meyer GmbH | Seelenhof 5 | 49100 Sch\u00f6nlingen |

Tel.: 0541/123456-1 | info@hansen-und-meyer.de

"

```
    }
  ],
  'attachments' => [
    {
      'content-type' => 'application/pdf',
      'name' => 'preisangebot.pdf',
      'size' => 44513
    },
    {
      'content-type' => 'application/pdf',
      'name' => 'agb.pdf',
      'size' => 63258
    }
  ],
  'header' => {
    'Cc' => [],
    'User-Agent' => 'Mozilla-Thunderbird 2.0.0.22 (X11/20090706)',
    'Subject' => "Re: Anfrage bez\u{c3}\x{bc}glich Partnerschaft",
    'MIME-Version' => '1.0',
    'In-Reply-To' => '<4AC1E7D2.6070302@web.de>',
    'Date' => 'Tue, 19 Jun 2012 13:22:38 +0200',
    'Bcc' => [],
    'Size' => 109490,
    'References' => '<4AC1E7D2.6070302@web.de>',
    'Message-ID' => '<4AC1EDFE.8070406@web.de>',
    'Content-Type' => 'multipart/mixed; boundary="-----
070101070508030007040901"',
    'To' => [
      'w.robert@robert-druck.de'
    ],
    'From' => 'info@hansen-und-meyer.de'
  }
};
```