



*Documentation*

---

## ***b2een REST API 1.0***

Last revised July 19<sup>th</sup>, 2011

## Revision history

Date	Author(s)	API version	Comments
2011-07-19	Simon Temple simon.temple@amalto.com	1.0	Document creation

## Table of contents

<b>1. Introduction .....</b>	<b>4</b>
1.1. Who Should Read this Guide .....	4
1.2. Required Knowledge and Skills .....	4
<b>2. Representation .....</b>	<b>4</b>
<b>3. Connectors .....</b>	<b>4</b>
<b>4. Authentication .....</b>	<b>4</b>
4.3. Authentication Tokens .....	5
4.4. User Roles .....	5
<b>5. Methods .....</b>	<b>6</b>
<b>6. Cross Origin Resource Sharing (CORS) .....</b>	<b>6</b>
<b>7. b2een Configuration Properties .....</b>	<b>7</b>
<b>8. Resources .....</b>	<b>8</b>
8.5. /ws/messages .....	9
8.5.1. Method: GET - Get a paged list of all messages .....	9
8.5.2. Method: PUT - Add a new message to the 'Draft' box .....	11
8.5.3. Method: POST - Add a new message to the 'Draft' box using an HTML form .....	13
8.5.4. Method: POST - Add a new message using a MIME encoded multipart/mixed entity .....	15
8.6. /ws/messages/{messageID} .....	17
8.6.1. Method: GET - Get message details, optionally mark as read .....	17
8.6.2. Method: DELETE - Delete a message (only if in the 'Draft' box) .....	18
8.6.3. Method: PUT - Update a message (only if in the 'Draft' box) .....	19
8.6.4. Method: POST - Add an attachment to a message in the 'Draft' box using an HTML form post .....	21
8.7. /ws/messages/{messageID}/{attachmentID} .....	23
8.7.1. Method: GET - Get the Attachment content .....	23
8.7.2. Method: POST - Add an attachment to a message in the 'Draft' box .....	24
8.7.3. Method: DELETE - Remove an attachment from a message (only if the message is in the 'Draft' box) ..	25
8.8. /ws/messages/feeds .....	26
8.8.1. Method: GET - List 'typical' feeds (auto generating securitytoken query parameters) .....	26
8.9. /ws/sectoken .....	27
8.9.1. Method: GET - Apply for token .....	27
8.10. /ws/communities .....	28
8.10.1. Method: GET - Get the list of all subscribed Communities .....	28
8.11. /ws/communities/{communityID}/partners .....	29
8.11.1. Method: GET - List all invited partners .....	29
8.12. /ws/plugins/{communityID}/resource/{name} .....	30
8.12.1. Method: GET - Get resource from Community plugin .....	30
<b>9. Request Failure and Errors .....</b>	<b>31</b>

# 1. Introduction

b2een REST API V1.0 provides web service control over b2een messages.

This API enables b2een to be used as a service server, with message creation, transmission and collection integration into third party applications and tools.

## 1.1. Who Should Read this Guide

---

This guide is for developers who are creating libraries to implement the b2een REST API. This audience can use this API reference to learn how the HTTP packets should look for particular requests.

## 1.2. Required Knowledge and Skills

---

Use of this guide assumes you are familiar with the following:

- XML (go to [W3 Schools XML Tutorial](#))
- Basic understanding of web services (go to [W3 Schools Web Services Tutorial](#))
- A programming language for consuming a web service and any related tools

# 2. Representation

The HTTP *Accept* header will be used to select response representations.

The header can be set to one of the following media types:

- text/html
- application/json
- application/rss+xml

In addition a query parameter: **alt**, can (when allowed) be use to select response representation.

# 3. Connectors

Both HTTP and HTTPS connectors are available. However because *headers* are transferred as 'clear text' when using HTTP and they can include authentication information, HTTPS is strongly recommended.

# 4. Authentication

Basic HTTP authentication is accepted for all resources.

HTTP status 401 will be returned if the authentication header is not present or authentication fails.

The user name supplied is that of a defined Portal user or '**admin**' in the case of stand alone installations.

The password supplied is that of a defined user or the WEB Interface password (in the case of stand alone installations).

## 4.3. Authentication Tokens

---

A '**securitytoken**' URL parameter can be used as an authentication mechanism when performing API calls.

A '**securitytoken**' is gained by accessing the `/ws/sectoken` resource. Access to this resource mandates Basic HTTP authentication.

A token has an optional expiry date and inherits the role of the user who is authenticated when accessing the **sectoken** resource.

Purpose:

1. Greater security when using clear text HTTP connectors.
2. Applications wishing to accept **application/rss+xml** resource representations (some RSS readers for example) may not support the basic authentication challenge and response mechanisms. Where this is the case, a **securitytoken** parameter can be used as a resource query parameter.

A **securitytoken** will fail resource authentication validation if:

1. The token has expired
2. The token has insufficient permissions for the resource/method being accessed
3. The password of the user has been changed since issuing the token (this process invalidates all issued tokens)

## 4.4. User Roles

---

b2een has three defined roles:

- administrator
- power
- standard

An **administrator** role is an unrestricted role and can use all available resources and user interface functions.

A **power** role allows unrestricted use of all service resources but has limited access to user interface functions.

A **standard** role allows 'read-only' access to resources and has a limited access to user interface functions.

A single user installation of b2een has a single user name of **admin**. This user is automatically associated with the **administrator** role.

Additional users can be created and assigned various roles and passwords.

To create users in b2een:

---

1. Add a 'users' folder to the \$USER/b2een folder
2. Add a file to the 'users' folder with the name of the user to create
3. Add the following properties to the username file created:
  - password: <password>
  - role: { administrator | power | standard }

## 5. Methods

This API uses the following HTTP methods:

- **GET** - Generally read-only operations (Some GET methods have optional 'update' side effects. See resource definitions for details).
- **POST** - Create operations
- **PUT** - Create/Update operations
- **DELETE** - Delete operations

Sometimes firewalls or restrictive runtime containers may disallow the use of one or more of these HTTP methods. Typically these are the **PUT** and/or **DELETE**.

To avoid such issues simply change the API HTTP method to a POST and add the following additional HTTP header to the request:

**X-HTTP-Method-Override:** <target method>

<Target method> being the method you would like to use but can't: PUT or DELETE.

For example:

If you wish to use the resource: /ws/messages/{messageID}/{attachmentID} to remove an attachment but cannot use the DELETE method, simply add the following request header and use POST instead:

**X-HTTP-Method-Override: DELETE**

## 6. Cross Origin Resource Sharing (CORS)

See <http://www.w3.org/TR/cors/>

If you use the REST API from JavaScript running in a browser and the b2een server is not running on the same server that served your script, techniques defined in the CORS standard are used to determine if cross-origin access is permitted.

A pre-flight request is made from the browser to the server to check if the actual request is allowed.

**The default behavior of b2een is to allow all cross-domain requests.** This can make installations vulnerable to a number of cross-domain exploits.

To secure your b2een installation, it is recommended that you set the following b2een Configuration Properties:

- ws.cors.allowhosts
- ws.cors.maxage

For additional information on b2een Configuration Properties, see next section.

## 7. b2een Configuration Properties

Access to the Configuration Properties is via the b2een Authentication panel displayed when b2een starts. Select "Options...", then "Advanced" and click on "Expert..." to edit properties.

The following b2een Configuration Properties are relevant to this API: all properties are optional.

### ❖ ws.port

The HTTP listen port of the service server. Typical value is 9080 however there is no default.

### ❖ ws.https.port

The HTTPS listen port of the service server. Typical value is 9083 however there is no default. A java key store is required when using HTTPS. The keystore must be called **jsse.jks** and placed in the \$USER/b2een/conf folder.

Details on how to create the keystore can be found here:

<http://download.oracle.com/javase/1.5.0/docs/guide/security/jsse/JSSERefGuide.html#CreateSimpleKeystore>

Note: If neither ws.port or ws.https.port are defined then the services server is **not** started.

The following \*.uri properties allow the static resources served as responses to service requests to be located externally. By default, all static resources are embedded in the b2een product and are served by the embedded WEB server. Therefore all URIs are embedded by default.

### ❖ ws.css.uri

Location of style sheets

### ❖ ws.image.uri

Location of images

### ❖ ws.script.uri

Location of Javascript

### ❖ ws.html.uri

Location of HTML template. Typically **footer.html** which is placed at the foot of each HTML representation generated.

### ❖ ws.rss.editor.email

The email address of the editor specified in the RSS feed.

### ❖ ws.rss.webmaster.email

The email address of the WEB master embedded in the RSS feed.

#### ❖ **ws.rss.feed.ttl**

The TTL attribute of a generated feed. See <http://cyber.law.harvard.edu/rss/rss.html> for more details.

#### ❖ **ws.css.names**

A comma separated list of stylesheet file names that will be located in an internal or external static files location and will be referred to from all HTML representations generated by this API. The file name **b2een.css** is used by default.

#### ❖ **ws.js.names**

A comma separated list of javascript file names that will be located in an internal or external static files location and will be referred to from all HTML representations generated by this API. There is no default script(s).

#### ❖ **ws.cors.allowhosts**

A comma separated list of hostnames that serve pages that require access to the b2een REST API. The hostnames provided must match those used to access the pages in a browser URL.

#### ❖ **ws.cors.maxage**

The number of seconds that a browser can cache a CORS pre-flight request result. This value has a default value of 1728000 (20 days). Note: this value may not directly influence how often a browser issues pre-flight requests.

## 8. Resources

*“In REST, stateless means that there is no client session data stored on the server. The server only records and manages the state of the resources it exposes.”* (see full article).

The b2een REST API V1.0 mainly provides services for manipulating b2een messages. Future releases may include control over other b2een functions such as community subscription and partner invitation.

The b2een REST API v1.0 resources have the following contexts:

- /ws/messages
- /ws/messages/{messageID}
- /ws/messages/{messageID}/{attachmentID}
- /ws/messages/feeds
  
- /ws/communities
- /ws/communities/{communityID}/partners
  
- /ws/plugins/{communityID}/resources/{name}

Support resource:

- /ws/sectoken

Each resource is now defined in detail with an accompanying example.

## 8.5. /ws/messages

All b2Een messages

### 8.5.1. Method: GET - Get a paged list of all messages

Roles: administrator, power, standard

#### ❖ Request Headers

Header	Value	Required
Accept	{text/html   application/json   application/rss+xml}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

#### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional
alt	{ rss   json }	Conditional
box	{in=1   out=2   error=4   sent=3   draft=6   deleted=5}	No
community	{community name}	No
partner	{partner name}	No
template	{document template name}	No
type	{document type name}	No
cdatefrom	{document create FROM date}	No - unless cdateto used
cdateto	{document create TO date}	No - unless cdatefrom used
read	{true   false}	No
page-size	{rows in a page} default: 50 (1024 for rss)	No
page	{page number} default: 1	No

## ❖ Example Request

```
curl -G --header "Accept: application/json" -uadmin:password  
'http://localhost:9080/ws/messages?page-size=1&page=3'
```

## ❖ Example Response

```
{ "MessageResource-array": [  
  {  
    "message": {  
      "messageID": "c423bc9a645c11e0a2780026bb634c76.arevatdtest",  
      "documentType": "Catalogue",  
      "documentTemplate": "AREVATD.Catalog",  
      "documentNumber": "c423bc9a645c11e0a2780026bb634c76",  
      "tradingPartnerUniqueID": "arevatdtest",  
      "tradingPartnerName": "Areva T&D TEST",  
      "tradingPartnerCompanyName": "Areva T&D",  
      "subject": "File 'Catalogue_99999_20100726_without-errors.xls' for 'Areva T&D TEST'",  
      "communityName": "Areva T&D TEST Community",  
      "technicalMessage": "Message: c423bc9a645c11e0a2780026bb634c76.arevatdtest to Areva  
T&D TEST [arevatdtest] of Areva T&D<br/>Status: SUCCESS. Delivered on Mon, 11 Apr  
2011 at 17:58:10 GMT+0100.",  
      "htmlMessage": "<br/>",  
      "createdDate": 1302541041156,  
      "sentDate": 1302541090333,  
      "box": 3,  
      "correlationID": "c423bc9a645c11e0a2780026bb634c76",  
      "sendingTries": 0,  
      "statusBitMask": 1,  
      "tradingPartnerSubAddress": "alstombuyer",  
      "subAddress": "admin",  
      "attachments": [  
        {  
          "id": "cd433f3b645c11e0a2780026bb634c76",  
          "messageId": "c423bc9a645c11e0a2780026bb634c76.arevatdtest",  
          "fileName": "b2een-8501585938422937532Catalogue_99999_20100726_without-errors.xls",  
          "size": 413696,  
          "embeddedData": true  
        }  
      ],  
      "properties": [  
      ]  
    },  
    "resourceLocation":  
    "http://localhost:9080/ws/messages/c423bc9a645c11e0a2780026bb634c76.arevatdtest?markR  
ead=true"  
  }  
]
```

## ❖ Notes

administrator and power roles will create resourceLocation elements with the markRead query parameter set to true.

Parameters: `cdatefrom` and `cdateto` are used to specify a document created date range. Both parameters must be specified if used at all. Parameter values must be supplied in `xsd:datetime` (ISO8601) format. See: <http://www.w3.org/TR/xmlschema-2/#dateTime> (3.2.7.1 *Lexical representation*)

If the *time* is not included as part of the `datetime` parameter a value of `00:00:00` is assumed.

All parameters that contain content unsuitable for inclusion in a URL should first be UR:Encode. Details can be found here: [http://www.w3schools.com/TAGS/ref\\_urlencode.asp](http://www.w3schools.com/TAGS/ref_urlencode.asp)

#### ❖ Working With The `statusBitMask` Response Attribute

The `statusBitMask` attribute can best be manipulated with bitwise operators (see [http://en.wikipedia.org/wiki/Bitwise\\_operation](http://en.wikipedia.org/wiki/Bitwise_operation)).

This single integer value contains zero or more flag values that apply to the message. In Java these values are defined as:

```
public final static int MESSAGE_STATUS_READ = 0x001;
public final static int MESSAGE_STATUS_INBOUND = 0x002;
public final static int MESSAGE_STATUS_COMPLETE = 0x004;
public final static int MESSAGE_STATUS_FLAGGED = 0x008;
public final static int MESSAGE_STATUS_BLUE = 0x010;
public final static int MESSAGE_STATUS_YELLOW = 0x020;
public final static int MESSAGE_STATUS_RED = 0x040;
public final static int MESSAGE_STATUS_GREEN = 0x080;
public final static int MESSAGE_STATUS_BLACK = 0x100;
```

## 8.5.2. Method: PUT - Add a new message to the 'Draft' box

Roles: administrator, power

#### ❖ Request Headers

Header	Value	Required
Accept	{application/json}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No
Content-Type	{text/plain   text/html}	Yes
x-b2een-subject		No
x-b2een-recipient-id	{b2een unique id}	Yes
x-b2een-community		Yes
x-b2een-recipient-subaddress		No
x-b2een-correlation-id		No

Header	Value	Required
x-b2een-document-template		No
x-b2een-document-type		No
x-b2een-document-number		No
x-b2een-technical-message		No

#### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional
subject		No
recipient-id	{b2een unique id}	Yes
community		Yes
recipient-subaddress		No
correlation-id		No
document-template		No
document-type		No
document-number		No
technical-message		No

The request content will be used as the message content. HTML formatted text can be supplied.

#### ❖ Response Headers

Header	Value	Required
x-b2een-message-id	String	Yes

#### ❖ Example Request

```
curl -X PUT -uadmin:password -H "Accept: application/json" -H "x-b2een-recipient-id: simont-win" -H "Content-Type: text/html" -d @./content.html -v 'http://localhost:9080/ws/messages/'
```

#### ❖ Example Response

```
< HTTP/1.1 200 OK
```

```

< Content-Length: 0
< Content-Type: application/json; charset=UTF-8
< Date: Thu, 11 Nov 2010 16:33:08 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< x-b2een-message-id: 53042992edb111df8f7b0026bb634c76.simont-win
<
* Connection #0 to host localhost left intact
* Closing connection #0

```

#### ❖ Notes

The resource will fail if the recipient has not been invited as a trading partner using the b2een UI.  
**Request parameter values will be used in preference to request headers if supplied.**

The use of request parameters may be 'safer' than using request headers; especially if sending none ASCII characters.

The ID of the new message is also returned as part of the response; encoded as a JSON string.

### 8.5.3. Method: POST - Add a new message to the 'Draft' box using an HTML form

Roles: administrator, power

#### ❖ Request Headers

Header	Value	Required
Accept	{text/html}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No
Content-Type	multipart/form-data	Yes

#### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional

#### ❖ Form Fields

Header	Value	Required
b2een-subject		Yes

Header	Value	Required
b2een-recipient-id	{b2een unique id}	Yes
b2een-community		Yes
b2een-recipient-subaddress		No
b2een-content		No
b2een-correlation-id		No
b2een-document-template		No
b2een-document-type		No
b2een-document-number		No
b2een-technical-message		No

#### ❖ Response Headers

Header	Value	Required
x-b2een-message-id	String	Yes

#### ❖ Example Request

```
curl -F b2een-recipient-id=simont-win -F b2een-subject=Test -F b2een-content=@./msg.txt -uadmin:password -H "Accept: text/html" -v 'http://localhost:9080/ws/messages'
```

#### ❖ curl Example Response

```
< HTTP/1.1 100 Continue
< HTTP/1.1 200 OK
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< Date: Mon, 15 Nov 2010 15:50:11 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< x-b2een-message-id: 01a7771df0d011dfaea00026bb634c76.simont-win
<
* Connection #0 to host localhost left intact
* Closing connection #0
```

#### ❖ Notes

The POST operation adds a message to the Drafts box using HTML forms. POST is an alternate form of PUT that enables browser-based uploads as a way of putting messages in the Drafts box. Parameters that are passed to PUT via HTTP Headers are instead passed as form fields to POST in the multipart/form-data encoded message body. The maximum size allowed for b2een-content is 1048576 bytes.

❖ **Content-Type(s)**

multipart/form-data[RFC1867]

The multipart/form-data content type is intended to allow information providers to express file upload requests uniformly, and to provide a MIME-compatible representation for file upload responses.

### 8.5.4. Method: POST - Add a new message using a MIME encoded multipart/mixed entity

**Note:** message content and attachments only. Request headers for all other message attributes. This style is often described as 'sending email via HTML'.

**Roles:** administrator only

❖ **Request Headers**

Header	Value	Required
Accept	{text/html}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No
Content-Type	multipart/mixed	Yes
x-b2een-box	{in=1   out=2   error=4   sent=3   draft=6   deleted=5} default 2	No
x-b2een-unqiue-id	{b2een unique id}	No
x-b2een-community-name	{b2een community id}	Yes
x-b2een-template-name		No
x-b2een-peer-name		No
x-b2een-peer-company-name		No
x-b2een-correlation-id		No
x-b2een-document-type		No
x-b2een-document-number		No
x-b2een-subject		No
x-b2een-username	Default: resource username	No

Header	Value	Required
x-b2een-peer-username		No

❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional

❖ MIME Attachments

Header	Value	Required
_message.html_	HTML message content	No
<Various named parts>	Additional content to be stored as message attachments	No

❖ Response Headers

Header	Value	Required
x-b2een-message-id	String	Yes

❖ Notes

This resource should be used with care. Because this resource allows many different values for attributes such as message box and username, it is possible to damage the internal message workflow of b2een.

The Content-type: header must also include the MIME boundary. For example:

```
Content-type: multipart/mixed boundary="----=_Part_22_1425160725.1290522669068"
```

## 8.6. /ws/messages/{messageID}

b2een message given ID

### 8.6.1. Method: GET - Get message details, optionally mark as read

Roles: administrator, power, standard

#### ❖ Request Headers

Header	Value	Required
Accept	{text/html   application/json}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

#### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional
markRead	{true   false}	No

#### ❖ Example Request

```
curl -uadmin:password -H "Accept: application/json" -v
'http://localhost:9080/ws/messages/f3140379d21711df99360026bb634c76'
```

#### ❖ Example Response

```
< HTTP/1.1 200 OK
< Content-Type: application/json; charset=UTF-8
< Date: Mon, 20 Dec 2010 14:20:04 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< Access-Control-Allow-Origin: *
< Transfer-Encoding: chunked
<
{"B2eenMessage": {
"messageID": "f3140379d21711df99360026bb634c76",
"documentType": "Files",
"documentTemplate": "FILES",
"documentNumber": "1286457882457",
"tradingPartnerUniqueID": "simont-win",
"tradingPartnerName": "simont-win",
"tradingPartnerCompanyName": "Amalto",
"subject": "test",
"communityName": "b2een B2B Community",
```

```

"technicalMessage": "Message: 1286457882457.simont-mac from simont-win [simont-win]
of Amalto<br/>Status: received on Thu Oct 07 14:36:54 BST 2010",
"htmlMessage": "test<br>",
"createdDate": 1286458614867,
"sentDate": -1,
"box": 1,
"correlationID": "1286457882457",
"sendingTries": -1,
"subAddress": "",
"read": true,
"inbound": true,
"attachments": [
],
"properties": [
]

```

#### ❖ Notes

standard role users cannot mark messages as read using this resource.

### 8.6.2. Method: DELETE - Delete a message (only if in the 'Draft' box)

Roles: administrator, power

#### ❖ Request Headers

Header	Value	Required
Accept	{application/json   text/html }	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

#### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional

#### ❖ Response Headers

Header	Value	Required

#### ❖ Example Request

```

curl -X DELETE -uadmin:password -H "Accept: application/json" -v
'http://localhost:9080/ws/messages/53042992edb111df8f7b0026bb634c76.simont-win'

```

### ❖ Example Response

```

< HTTP/1.1 200 OK
< Content-Length: 0
< Content-Type: application/json; charset=UTF-8
< Date: Fri, 12 Nov 2010 10:04:21 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
<
* Connection #0 to host localhost left intact
* Closing connection #0

```

### ❖ Notes

This method is typically used to remove a message from the 'Draft' box.

## 8.6.3. Method: PUT - Update a message (only if in the 'Draft' box)

Roles: administrator, power

### ❖ Request Headers

Header	Value	Required
Accept	{application/json}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No
Content-Type	{text/plain   text/html}	Yes
x-b2een-subject		No
x-b2een-recipient-id	{b2een unique id}	No
x-b2een-community		Conditional
x-b2een.recipient-subaddress		No
x-b2een-box	{in=1   out=2   error=4   sent=3   draft=6   deleted=5} default:2	No
x-b2een-correlation-id		No
x-b2een-document-template		No
x-b2een-document-type		No
x-b2een-document-number		No

Header	Value	Required
x-b2een-technical-message		No

❖ Request Parameters

Parameter	Value	Required
appendcontent	{true   false}	No
securitytoken	{valid authentication token}	Conditional
subject		No
recipient-id	{b2een unique id}	No
community		No
recipient-subaddress		No
box	{in=1   out=2   error=4   sent=3   draft=6   deleted=5} default:2	No
correlation-id		No
document-template		No
document-type		No
document-number		No
technical-message		No

The request content will be used as the message content. HTML formatted text can be supplied. Content will replace existing content unless **appendcontent=true**

❖ Response Headers

Header	Value	Required
x-b2een-message-id	String	Yes

❖ Example Request

```
curl -X PUT -uadmin:password -H "Accept: application/json" -H "Content-Type: text/plain" -v -d "Hello" 'http://localhost:9080/ws/messages/01a7771df0d011dfaea00026bb634c76.simont-win?appendcontent=true'
```

❖ Example Response

```
< HTTP/1.1 200 OK
```

```

< Content-Length: 0
< Content-Type: application/json; charset=UTF-8
< Date: Mon, 15 Nov 2010 16:23:34 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< x-b2een-message-id: 01a7771df0d011dfa00026bb634c76.simont-win
<
* Connection #0 to host localhost left intact
* Closing connection #0

```

❖ **Notes**

This method is typically used to 'move' a message from the Draft box to the Out box where it will be sent to its destination.

The message must exist and be 'owned' by the resource user.

**Request parameter values will be used in preference to request headers if supplied.**

The use of request parameters may be 'safer' than using request headers; especially if sending non-ASCII characters.

The ID of the updated message is also returned as part of the response; encoded as a JSON string.

### 8.6.4. Method: POST - Add an attachment to a message in the 'Draft' box using an HTML form post

Roles: administrator, power

❖ **Request Headers**

Header	Value	Required
Accept	{text/html}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No
Content-Type	multipart/form-data	Yes

❖ **Request Parameters**

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional

❖ **Form Fields**

Header	Value	Required
b2een-att-name		Yes

Header	Value	Required
b2een-att-content		Yes

❖ **Response Headers**

Header	Value	Required
x-b2een-attachment-id	String	Yes

❖ **Example Request**

```
curl -F "b2een-att-name=att.txt" -F "b2een-att-content=1234567890" -uadmin:password -H "Accept: text/html" -v 'http://localhost:9080/ws/messages/01a7771df0d011dfa00026bb634c76.simont-win'
```

❖ **Example Response**

```
< HTTP/1.1 100 Continue
< HTTP/1.1 200 OK
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< Date: Tue, 16 Nov 2010 09:58:03 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< x-b2een-attachment-id: 0087165cf16811df8d010026bb634c76
<
* Connection #0 to host localhost left intact
* Closing connection #0
```

❖ **Notes**

The POST operation adds an attachment to message in the Drafts box using HTML forms. This is an alternate form of POST (see `ws/message{messageId}/{attachmentId}`) that enables browser-based uploads as a way of adding attachments to messages in the Drafts box. Parameters are passed as form fields to POST in the multipart/form-data encoded message body.

## 8.7. /ws/messages/{messageID}/{attachmentID}

b2een message attachment given message ID and attachment ID

### 8.7.1. Method: GET - Get the Attachment content

Roles: administrator, power, standard

#### ❖ Request Headers

Header	Value	Required
Accept	*/*	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

#### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional
inline	Force Content-Disposition to inline. The default is attachment.	No

#### ❖ Example Request

```
curl -G -uadmin:password -v -o attachment.dat
'http://localhost:9080/ws/messages/c4612dcee0f011dfa3ab0026bb634c76/c461f11fe0f011dfa3ab0026bb634c76'
```

#### ❖ Example Response

```
< Content-Length: 2719744
< Content-Type: image/jpeg; charset=UTF-8
< Last-Modified: Thu, 28 Oct 2010 12:59:34 GMT
< Content-Disposition: attachment; filename=DSCN3622.JPG
< Date: Thu, 28 Oct 2010 12:59:34 GMT
< Accept-Ranges: bytes
< Server: b2een: 3.0.0_101-2010/09/06T15:18
<
{ [data not shown]
```

#### ❖ Notes

None.

## 8.7.2. Method: POST - Add an attachment to a message in the 'Draft' box

The {attachmentId} specifies the name of the attachment being posted.

Roles: administrator, power

### ❖ Request Headers

Header	Value	Required
Accept	{application/json}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional

### ❖ Response Headers

Header	Value	Required
x-b2een-attachment-id	String	Yes

### ❖ Example Request

```
curl -X POST -d @./myfile.txt -uadmin:password -H "Accept: application/json" -v
'http://localhost:9080/ws/messages/01a7771df0d011dfaea00026bb634c76.simont-
win/myfile.txt'
```

### ❖ Example Response

```
< HTTP/1.1 200 OK
< Content-Length: 0
< Content-Type: application/json; charset=UTF-8
< Date: Tue, 16 Nov 2010 11:40:17 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< x-b2een-attachment-id: 478b4a82f17611dfbc4b0026bb634c76
<
* Connection #0 to host localhost left intact
* Closing connection #0
```

### ❖ Notes

None.

### 8.7.3. Method: DELETE - Remove an attachment from a message (only if the message is in the 'Draft' box)

Roles: administrator, power

#### ❖ Request Headers

Header	Value	Required
Accept	{application/json   test/html}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

#### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional

#### ❖ Example Request

```
curl -X DELETE -uadmin:password -H "Accept: application/json" -v  
'http://localhost:9080/ws/messages/01a7771df0d011dfa00026bb634c76.simont-  
win/478b4a82f17611dfbc4b0026bb634c76'
```

#### ❖ Example Response

```
< HTTP/1.1 200 OK  
< Content-Length: 0  
< Content-Type: application/json; charset=UTF-8  
< Date: Tue, 16 Nov 2010 12:05:44 GMT  
< Accept-Ranges: bytes  
< Server: b2een 3.0.0_101-2010/09/06T15:18  
<  
* Connection #0 to host localhost left intact  
* Closing connection #0
```

#### ❖ Notes

None.

## 8.8. /ws/messages/feeds

Generated feeds access.

### 8.8.1. Method: GET - List 'typical' feeds (auto generating securitytoken query parameters)

Roles: administrator, power, standard

#### ❖ Request Headers

Header	Value	Required
Accepts	{text/html   application/json}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

#### ❖ Request Parameters

Parameter	Value	Required
validity	{days} default: 0 - forever	No
securitytoken	{valid authentication token}	Conditional

#### ❖ Example Request

```
curl -uadmin:password -H "Accept: application/json" -v
'http://localhost:9080/ws/messages/feeds'
```

#### ❖ Example Response

```
< HTTP/1.1 200 OK
< Content-Type: application/json; charset=UTF-8
< Date: Tue, 23 Nov 2010 08:22:51 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< Transfer-Encoding: chunked
<
* Connection #0 to host localhost left intact
* Closing connection #0
{"list":[{"FeedResource":{"ResourceURI":"http://localhost:9080/ws/messages?read=false&box=1&alt=rss&page-size=1024&securitytoken=745x5wvtbdwftyvejcy24c2tj7wtp16btbv5ajuco7omkac7qnmybb5tpnltnwnf"}, {"ResourceURI":"http://localhost:9080/ws/messages?read=false&box=4&alt=rss&page-size=1024&securitytoken=745x5wvtbdwftyvejcy24c2tj7wtp16btbv5ajuco7omkac7qnewqkzlkdwjcyg"}, {"ResourceURI":"http://localhost:9080/ws/messages?alt=rss&page-size=1024&securitytoken=745x5wvtbdwftyvejcy24c2tj7wtp16btbv5ajuco7omkac7qnmuhxzzso5q"
```

```
tdb"}}}}}
```

#### ❖ Notes

Feed URIs are:

- Inbox/Unread -> All unread messages in the Inbox.
- Errors/Unread -> All unread messages in the Errorbox.
- All (max. first 1024 messages only) -> All messages in all boxes.

## 8.9. /ws/sectoken

Access authentication security token

### 8.9.1. Method: GET - Apply for token

Roles: administrator, power, standard

#### ❖ Request Headers

Header	Value	Required
Accepts	{text/html   application/json}	Yes
Authentication: basic	{base64 encoded username:password}	Yes
Accept-Language	{acceptable language for response}	No

#### ❖ Request Parameters

Parameter	Value	Required
validity	{days} default: 0 - forever	No

#### ❖ Example Request

```
curl -G -H "Accept: application/json" -uadmin:password -v  
'http://localhost:9080/ws/sectoken'
```

#### ❖ Example Response

```
< HTTP/1.1 200 OK  
< Content-Type: application/json; charset=UTF-8  
< Date: Mon, 01 Nov 2010 11:40:52 GMT  
< Accept-Ranges: bytes  
< Server: b2een: 3.0.0_101-2010/09/06T15:18  
< Transfer-Encoding: chunked  
<  
* Connection #0 to host localhost left intact  
* Closing connection #0  
{  
  "list": [{"string": "745x5wvtbdwftyvejcy24c2tj7wtp16btbv5albmsehh3qmwrb424zyprhddi"}]  
}
```

❖ **Notes**

It is important to note that you cannot use a security token to authenticate yourself when accessing this resource.

## 8.10. /ws/communities

---

All b2een communities currently subscribed to

### 8.10.1. Method: GET - Get the list of all subscribed Communities

Roles: administrator, power, standard

❖ **Request Headers**

Header	Value	Required
Accept	{application/json}	Conditional
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

❖ **Request Parameters**

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional

❖ **Example Request**

```
curl -uadmin:password -H "Accept: application/json" -v 'http://localhost:9080/ws/communities'
```

❖ **Example Response**

```
< HTTP/1.1 200 OK
< Content-Type: application/json; charset=UTF-8
< Date: Tue, 23 Nov 2010 10:05:17 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< Transfer-Encoding: chunked
<
* Connection #0 to host localhost left intact
* Closing connection #0
{"list":[{"FeedResource":{"communityId":"Amalto B2B
Community","parentCommunityId":"## NPG ##","status":4}, {"communityId":"Atos
Origin","parentCommunityId":"## NPG
##","status":3}, {"communityId":"NetPeerGroup","status":4}, {"communityId":"Total R&M
Back-office Community","parentCommunityId":"## NPG
##","status":3}, {"communityId":"b2een B2B Community","parentCommunityId":"Amalto B2B
Community","status":3}]}}
```

#### ❖ Notes

Status values are:

- 1: An error occurred and is fatal
- 0: An error occurred and can hopefully be recovered without a restart
- 1: The initial state of the community or of an unsubscribed community.
- 2: Subscription was successful but peer voluntarily decided to logout/not connect
- 3: The Peer is logged and connecting to the community
- 4: The peer is logged and connected

## 8.11. /ws/communities/{communityID}/partners

b2een partners given community ID.

### 8.11.1. Method: GET - List all invited partners

Roles: administrator, power, standard

#### ❖ Request Headers

Header	Value	Required
Accept	{text/html   application/json}	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

#### ❖ Request Parameters

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional
alias	{partner subscription alias}	No

#### ❖ Example Request

```
curl -uadmin:password -H "Accept: application/json" -v
'http://localhost:9080/ws/communities/Atos+Origin/partners'
```

#### ❖ Example Response

```
< HTTP/1.1 200 OK
< Content-Type: application/json; charset=UTF-8
< Date: Wed, 08 Dec 2010 15:13:35 GMT
< Accept-Ranges: bytes
< Server: b2een 3.0.0_101-2010/09/06T15:18
< Access-Control-Allow-Origin: *
< Transfer-Encoding: chunked
<
{"PartnerResource-array": [
  {
```

```

    "lastPresence": -1,
    "status": 1,
    "bio": {
      "companyName": "Atos Origin",
      "description": "Atos Origin",
      "street1": "",
      "street2": "",
      "city": "Paris",
      "state": "",
      "zipCode": "",
      "country": "France",
      "uniqueID": "atos",
      "properties": [
      ],
      "adminPassword": "",
      "agreementNumber": "",
      "type": 1
    }
  }
}
* Connection #0 to host localhost left intact
* Closing connection #0

```

❖ **Notes**

Remember a CommunityID must be URL Encoded to be included as part of a resource path. Details can be found here: [http://www.w3schools.com/TAGS/ref\\_urlencode.asp](http://www.w3schools.com/TAGS/ref_urlencode.asp)

If the parameter **alias** is used, only those partners within the selected community who have an alias which matches that given will be returned.

## 8.12. /ws/plugins/{communityID}/resource/{name}

b2een community plugin resource given Community ID and resource name

### 8.12.1. Method: GET - Get resource from Community plugin

Roles: administrator, power, standard

❖ **Request Headers**

Header	Value	Required
Accept	*/*	Yes
Authentication: basic	{base64 encoded username:password}	Conditional
Accept-Language	{acceptable language for response}	No

❖ **Request Parameters**

Parameter	Value	Required
-----------	-------	----------

Parameter	Value	Required
securitytoken	{valid authentication token}	Conditional
inline	Force Content-Disposition to inline. The default is attachment.	No

#### ❖ Example Request

```
curl -G -uadmin:password -v -o attachment.dat
'http://localhost:9080/ws/plugins/Atos+Origin/resources/mobilerules.js'
```

#### ❖ Example Response

```
< Content-Length: 2719744
< Content-Type: plain/txt; charset=UTF-8
< Last-Modified: Thu, 28 Oct 2010 12:59:34 GMT
< Content-Disposition: attachment; filename=mobilerules.js
< Date: Thu, 28 Oct 2010 12:59:34 GMT
< Accept-Ranges: bytes
< Server: b2een: 3.0.0_101-2010/09/06T15:18
<
{ [data not shown]
```

#### ❖ Notes

Allows access to any file in the *IMETA-INF/resources* folder of a community plugin JAR file by the community plugins class loader.

Remember a CommunityID and resource name must be URL Encoded to be included as part of a resource path. Details can be found here: [http://www.w3schools.com/TAGS/ref\\_urlencode.asp](http://www.w3schools.com/TAGS/ref_urlencode.asp)

## 9. Request Failure and Errors

When there is an error, the response header contains:

- Content-Type: As requested by the Accept header of the request otherwise text/html is assumed
- An appropriate 3xx, 4xx, or 5xx HTTP status code

The body or the response also contains information about the error. The following text/html sample response shows the structure of response elements common to all service error responses.

```
Service Error : 400
Message      Invalid parameter value given for page: ONE
Error Type   B2Exception
Resource     /messages?page=ONE
```

If Accept: application/json was requested the error response is encapsulated in an **ErrorResource** object.

ErrorType	Description	HTTP Status Code
-----------	-------------	------------------

ErrorType	Description	HTTP Status Code
B2Exception	Request Parameter(s) or operation invalid	400
B2Exception	Resource not found	404
B2Exception	Operation on resource is disallowed	403
<various>	Internal service error	500
B2Exception	Unauthorized	401