MALMÖ
UNIVERSITET

Bulletin Board API: Documentation

Introduction

This is the documentation for Version 1 of the Bulletin Board API (BBAPI.1)

Resource Formats

The default resource format is JSON (application/json).

Authentication

BBAPI.1 runs on a username/password authentication protocol.
Only username/password pairs valid for Chunks are valid.
Authentication data is sent with parameters (GET) or in the body of the request (POST, PATCH, DELETE).

Prefix

All End Points (i.e. resources) need to be prefixed with:
https://teaching.maumt.se/apis/bulletin_board/v1/

Resource

BBAPI.1 serves objects with this structure:

```
{
  response_feedback: string,      text-based feedback from the server
  messages: array of Messages,
  users: array of Users
}
```

…where Messages are objects with the structure:

```
{
  message_id: int                 message id
  sender: string,                 username of the sender
  receiver: string,               username of the receiver
  content: string,                content of the message
  timestamp: string,              the date of dispatch, with format “2022-09-12 09:05:03”
  edited_timestamp: string,       “0” if message has not been edited, else date of the latest edition.
}
```

…and Users are objects with the structure:

```
{
  name: string                    username
}
```

The array of Messages will be ordered by timestamp, decreasing (from latest to earliest).
The array of Users will be ordered alphabetically.
Examples

```
prefix?un=X&pw=Y&sender_name=Z            serves all messages where X is receiver and Z is sender
```

Accessing users from the bulletin

A list of all users in the bulletin is located in the resource:

```
prefix?un=X&pw=Y&all_users                serves all users in the DB
```

A list of all the users that have sent or received a message to or from a specific user is located in the resource:

```
prefix?un=X&pw=Y&connections              serves all users that have a message to or from un
```

Accessing messages from the bulletin

Messages from the bulletin can only be accessed with GET requests. Required GET-parameters:

un: A valid Chunks username
pw: A valid Chunks password
sender or receiver: one and only one of them must be set and be a valid Chunks username

If *sender* is set, BBAPI.1 will return messages where *un* is the receiver and *sender* the sender.
If *receiver* is set, BBAPI.1 will return messages where *un* is the sender and *receiver* the receiver.
In all cases, BBAPI.1 will only send the latest 10 messages.
If there is no user in the DB with the name *sender* or *receiver*, BBAPI.1 responds with a 404 code.

Examples

```
prefix?un=X&pw=Y&sender=Z         serves all messages where X is receiver and Z is sender
```

Posting Messages

Messages are posted with a POST request. Note that the message will have sender = username (from username/password).
The request must have the header "Content-type": "application/json; charset=UTF-8". The body of the request must be a JSON-object with the structure:

```
{
  un: string,                     username of the sender
  pw: string,                     password of the sender
  receiver: string,               username of the receiver
  content: string,                content of the message
}
```

The response from a POST-request has the form:

```
{
  response_feedback: string,      text-based feedback from the server
  messages:
    [
      {
         message_id: int
         sender: string,
         receiver: string,
         content: string,
         timestamp: int,
         edited_timestamp: int,
      }
    ]
}
```

…where messages is an array of only one object: the message just posted.

Possible responses

	status code	response_feedback	notes
Successful posting	201	OK	Assigned id is included in response
Invalid username/password pair	400	Invalid credentials	No changes to DB
User not found	404	User not found	There is no user with the name passed in "receiver"

Patching Messages

Existing messages in the Bulletin Board can be edited with a PATCH-request. Note that only messages with sender = username (from username/password) can be edited.
The request must have the header "Content-type": "application/json; charset=UTF-8". The body of the request must be a JSON-object with the structure:

```
{
  un: string,                     username of the sender
  pw: string,                     password of the sender
  message_id: int,                id of the message to be edited
  content: string,                new content of the message
}
```

The response from a PATCH-request has the form:

```
{
  response_feedback: string,      text-based feedback from the server
  messages:
    [
      {
         message_id: int
         sender: string,
         receiver: string,
         content: string,
         timestamp: int,
         edited_timestamp: int,
      }
    ]
}
```

…where messages is an array of only one object: the message just patched.

Possible responses

	status code	response_feedback	notes
Successful update	200	OK	key "content" in response object will contain the updated content
Invalid username/password pair	400	Invalid credentials	No changes to DB
Sender has no message with that message_id	404	Message not found	No changes to DB

Deleting Messages

Existing messages in the Bulletin Board can be removed with a DELETE-request. Note that only messages with sender = username (from username/password) can be removed.
The request must have the header "Content-type": "application/json; charset=UTF-8". The body of the request must be a JSON-object with the structure:

```
{
  un: string,                     username of the sender
  pw: string,                     password of the sender
  message_id: int,                id of the message to be removed
}
```

The response from a DELETE-request has the form:

```
{
  response_feedback: string,      text-based feedback from the server
  messages:
    [
      {
         message_id: int
         sender: string,
         receiver: string,
         content: string,
         timestamp: int,
         edited_timestamp: int,
      }
    ]
}
```

…where messages is an array of only one object: the message just removed.

Possible responses

	status code	response_feedback	notes
Successful update	200	OK	The removed message is sent back in the response but the message does not exist any longer in the DB.
Invalid username/password pair	400	Invalid credentials	No changes to DB
Sender has no message with that message_id	404	Message not found	No changes to DB