

## Direct Messages Microservice Documentation

This first section covers design justification and the “schema”. Please scroll halfway through for API documentation.

### Explanation

I am not using any LSI, so the 10GB size limit of a partition will not be a problem, as the limit only applies to tables with one or more LSI.

Using recipient ID as a partition key is relatively high cardinality. If issues arise we could consider a partition key involving appending senderID to recipient ID for better write uniformity, although this means needing a GSI for listDirectMessagesFor(). I assumed listDirectMessagesFor() is the primary access pattern, so I built the table around it.

I use msgid as the sort key, where msgid is the sender’s username appended to the timestamp. This allows the messages to remain sorted while avoiding conflicts. Ideally I would use fancy a snowflake value like Twitter does. If users hypothetically manage to send two messages in very close succession, then the first write wins.

Given the required operations, I have determined the following access patterns.

**With PK = “to” SK = “msgid”**

Access Pattern	Table/GSI	Key Conditions
listDirectMessagesFor() List direct messages for a given recipient username.	Table	<b>Partition Key: “to” = recipient username</b> <b>Sort Key: msgid</b>
listRepliesTo() Before listing replies, query the original msgid to ensure the message exists.	GSI	<b>Primary Key: “msgid” = original msgid</b>
listRepliesTo() List message replies for a given message Id	GSI	<b>Partition Key: “in_reply_to” = msgid</b> <b>Sort Key: “msgid”</b>

### Attributes

To save space, short attribute names are used. Here is an explanation of their meanings:

t = to

fr = from

irt = in-reply-to

ts = timestamp

msg = message

qro = quick-reply-options

mid = msgid

**Table: Messages****Partition Key: t****Sort Key: mld**

Primary Key		Attributes				
Partition Key: t	Sort Key: mld	fr	irt	ts	msg	gro
John	FROM#Mary					['I'm interested.', 'No thank you.']
	2020-11-27T10:20:30.45+01:00Mary	Mary	none	2020-11-27T10:20:30.45+01:00	"Hello John."	
	2020-11-27T10:21:30.45+01:00Mary	Mary	none	2020-11-27T10:21:30.45+01:00	"You're invited to the event on Sunday!"	
Mary	FROM#John					[ ]
	2020-11-27T10:29:30.45+01:00John	John	2020-11-27T10:21:30.45+01:00Mary	2020-11-27T10:29:30.45+01:00	"I'm interested."	

Since the user must see the quick reply options to select one, they will have that information on the client side. Rather than handling quick replies on the back end, I expect the application to compose the quick reply as a regular message on the client side.

Ideally, I would only store two lists of quick replies per conversation-- one item for each user as shown above. However, I did not have time to implement this, so I am storing quick replies (if exists) with each message.

**Global Search Index: replies****PK: irt (in-reply-to), SK: mld (messageld)****Projection: All attributes**

Primary Key		Projected Attributes				
Partition Key: fr	Sort Key: mld	t	irt	ts	msg	qro
John	2020-11-27T10:29:30.45+01:00John	Mary	2020-11-27T10:21:30.45+01:00Mary	2020-11-27T10:29:30.45+01:00	"I'm interested."	
Mary	2020-11-27T10:20:30.45+01:00Mary	John	none	2020-11-27T10:20:30.45+01:00	"Hello John."	
Mary	2020-11-27T10:21:30.45+01:00Mary	John	none	2020-11-27T10:21:30.45+01:00	"You're invited to the event on Sunday!"	['I'm interested.', 'No thank you.']

**Global Search Index: messagelds****PK: mld (messageld), SK: None****Projection: t (to)**

The current access patterns do not call for projecting all the attributes here. Just using this to check if a previous message exists and who the sender was.

Primary Key	Projected Attributes
Partition Key: mld	t
2020-11-27T10:29:30.45+01:00John	Mary
2020-11-27T10:20:30.45+01:00Mary	John
2020-11-27T10:21:30.45+01:00Mary	John

**sendMessage(to, from, message, quickReplies=None)**

Sends a direct message to a user. The quickReplies parameter is optional.

**Gateway API Endpoint:** /api/v1/dms

**HTTP Request Method:** POST

**Request JSON:**

```
{
  "to": <string>
  "message": <string>
  ["quickreplies": <list<string>>]
}
```

**Other Info:**

- "from" will be gathered from the authentication submitted using basic auth.

**On Success:**

- 201 Created

**On Failure:**

- 500 Internal Server Error if put\_item fails
- 401 Unauthorized if user specifies a "from" value different from their authentication details.

### replyToDirectMessage(messageId, reply)

Reply to a particular message (perhaps displaying the original message as a quote on the application). Replying with a quick index is not supported. The client application must compose the quick reply as a regular message on the client side. The client application should have this information when messages are sent.

**Gateway API Endpoint:** /api/v1/dms/<string:messageId>/replies

**HTTP Request Method:** POST

**Request JSON:**

```
{
    "message": <string>
}
```

**Other Info:**

- "from" will be gathered from the authentication submitted using basic auth.
- "to" will be deduced from the messageId that this message is replying to.

**On Success:**

- 201 Created

**On Failure:**

- 401 Unauthorized if user tries to specify a "from" different from their authentication details.
- 404 Not Found if the messageId has no message associated with it.
- 500 Internal Server Error if put\_item fails.

### **listDirectMessagesFor(username)**

Get all the direct messages sent to a user. An optional timestamp may be provided to limit how far back the retrieval goes.

**Gateway API Endpoint:** /api/v1/dms

**HTTP Request Method:** GET

**Request JSON:**

```
{  
  ["timestamp": <datetime>]  
}
```

**Other Info:**

- "to" will be gathered from the authentication submitted using basic auth.

**On Success:**

- 200 OK

**On Failure:**

- 401 Unauthorized if user tries to specify a "to" different from their authentication details.
- 500 Internal Server Error if the query fails. (Still responds with 200 if query results are empty).

## listRepliesTo(messageId)

Lists the replies to a DM, listing all the messages “quoting” this message.

**Gateway API Endpoint:** /api/v1/dms/<string: messageId>/replies

**HTTP Request Method:** GET

**Request JSON:**

```
{  
    ["timestamp": <datetime>]  
}
```

**Other Info:**

- “to” will be gathered from the authentication submitted using basic auth.

**On Success:**

- 200 OK

**On Failure:**

- 500 Internal Server Error if the query fails. (Still responds with 200 if query results are empty).