

Design and implement a production-grade horizontally scalable REST service that implements a simple ephemeral text message service. The service should implement a hot/cold storage system where expired messages are stored separately from unexpired messages. The service has three RESTful endpoints for accessing data:

POST /chats

Creates a new text message for passed in username.

Request body properties

Name	Type	Description	Required	Default Value
username	String	The recipient of the message	Y	n/a
text	String	The content of the message	Y	n/a
timeout	Integer	The number of seconds the message should live before expiring	N	60

Response

A success response will include:

- a status code of **201 Created**
- a body with a JSON object containing the id of the newly-created message, for example:

```
{ "id": 5488 }
```

GET /chats/:id

Returns the message object for the given id. This service can return both expired and unexpired messages.

Response

A success response will include:

- a JSON response which contains the username, text, and expiration date.

Example response body (formatted for readability)

```
{
  "username": "paulrad",
  "text": "This is a message"
  "expiration_date": "2015-08-12 06:22:52"
}
```

GET /chats/:username

Returns a list of all unexpired texts for the user with the given username. Any texts that are received are then expired.

Response

A success response will include:

- a JSON array of messages, each of which contains the message's ID and

text. Example response body (formatted for readability)

```
[
  {
    "id": 345,
    "text": "This is a message"
  },
  {
    "id": 95958,
    "text": "This is also a message"
  }
]
```

Please include tests. You may use any programming language, framework, and storage solution you wish.