

RestChat Overview

I implemented a minimal chat functionality using Rest Services. The implemented feature is called “unread messages”.

Requirements

- Project is written with Eclipse Kepler
- Minimum: Java 6 (the project was developed in a Java 7 environment)
- Minimum: Glassfish 3 (the project was deployed using Glassfish 4)

Sending messages:

Using this URL

<http://localhost:8080/RESTfulChat/reso/rest/sendMsg?from=1&to=2&msg=HelloWorld&key=123> a message can be sent to a certain user. It remains in an “unread” state until is retrieved by a user.

The functionality is implemented using a HashMap. The key represents the recipients’ address and the mapped value is a list of MessagePacks.

MessagePacks represents a class that keeps together the sender, timestamp and the message.

Retrieving messages:

When the unread messages are retrieved by this URL :

<http://localhost:8080/RESTfulChat/reso/rest/getMsgs?me=2&key=456> the user gets all the unread messages in a JSON format.

The next step should be the transfer of the messages in an archive (or conversation history) because the messages in the HashMap are there only temporarily.

Security

The security is assured using the “key” query parameter. Key can be a hidden input value or the session ID or it could be both. Also, data transfer should be made using HTTPS.

Error Codes/Response

In any case, right HTTP codes should be returned (with respect to the family 2xx, 3xx, 4xx, 5xx).

Future implementation:

Respecting the UML Diagram, MessageService, ConferenceService and UserService should be implemented.

Nevertheless the database connection should be implemented.

Better solution

From my point of view, a better solution would be the usage of WebSockets protocol. Websockets provides full duplex communication using channels. In order to assure “live” or “real-time” conversation rather than using multiple AJAX requests it can be used channels in order to communicate.