**FOLLOWING ARE THE HEADERS AND BODY FORMAT OF SOMEIMPORTANT HTTTP CALLS AND STOMP FRAMES:**

* **for SUBSCRIBE frame in default destination subscription:  
  headers: {  
   id: <not needed>,  
   destination: <default destination>,  
   ack: “client-individual”,  
   user-id: <id of the current user>  
  }**
* **for SUBSCRIBE frame in chat specific destination subscription:  
  header: {  
   id: <not needed>,  
   destination: <chat destination>,  
   ack: “client-individual”,  
   user-id: <id of the user>,  
   chat-id: <id of the chat>  
  }**
* **for SEND frame in sendDeliveryAck():  
  header: {  
   destination: <default destination of the msg sender>,  
   content-type: “application-json”,  
   content-length: <length of the body in bytes>,  
   chat-id: <id of the chat>,  
   msg-id: <id of the msg>,  
   type: “delivery”  
  }  
  body: { //a ReceiverObject  
   ref: UserObject  
   timestamp: <received timestamp>,  
   status: 1  
  }**
* **for SEND frame in sendSeenAck():  
  header: {  
   destination: <default destination of the msg sender>,  
   content-type: “application-json”,  
   content-length: <length of the body in bytes>,  
   chat-id: <id of the chat>,  
   msg-id: <id of the msg>,  
   type: “seen”  
  }  
  body: { //a ReceiverObject  
   ref: UserObject  
   timestamp: <received timestamp>,  
   status: 2  
  }**
* **for SEND frame in send():  
  header: {  
   destination: <chat specific destination>,  
   content-type: “application-json”,  
   content-length: <length of the body in bytes>,  
   chat-id: <id of the chat>,  
   receipt: <id of the msg>  
  }  
  body: { //a MessageObject  
   id: <msg id>,  
   sender: { //a SenderObject  
   ref: UserObject  
   timestamp: <sent timestamp>  
   },  
   receiver: null,  
   textData: <raw text data>,  
   fileData: null,  
   noOfSeen: 0,  
   noOfDeliveries: 0,  
   deliveryStatus: -1  
  }**
* **for an individual user object request:  
  url: /messenger/user  
  method: GET  
  header: {  
   Authorization: <jwt token>,  
   u\_id : <user id>,  
  }  
  query: {  
   id : <target user id>  
  }  
  body: none  
  response🡪  
  status:   
   OK 🡪 successful response  
   BAD\_REQUEST 🡪 target user id not found  
   UNAUTHORIZED 🡪 not mutual following  
   INTERNAL\_SERVER\_ERROR 🡪 server error  
  body:  
   onSuccess 🡪 UserObject in JSON  
   onError 🡪 ErrorObject**
* **for an individual ChatObject request:  
  url: /messenger/chat  
  method: GET  
  header: {  
   Authorization: <jwt token>,  
   u\_id : <user id>,  
  }  
  query: {  
   id : <target chat id>  
  }  
  body: none  
  response🡪  
  status:  
   OK 🡪 successful response  
   BAD\_REQUEST 🡪 target chat id not found  
   UNAUTHORIZED 🡪 not a receipient  
   INTERNAL\_SERVER\_ERROR 🡪 server error  
  body:  
   onSuccess 🡪  
   {// a ChatObject  
   id : <chat id>,  
   title : <title of the chat>,  
   picture : <picture of the chat>,  
   destination : <destination fo the chat>,  
   totalNoOfMsg : <total no of msg in the chat>,  
   deliveredNoOfMsg : <total no of delivered of msg>,  
   seenNoOfMsg : <total seen no of msg>,  
   messages: [ //list of MessageObject  
   {  
   id : <id of the msg>,  
   sender: {  
   ref: UserObject,  
   timestamp: <sent timestamp>  
   },  
   receivers: [ // list of ReceiverObject  
   {  
   ref: UserObject,  
   timestamp: <received timestamp>,  
   status: 1(delivered)/ 2(seen)  
   }  
   ],  
   textData: <raw text string>,  
   fileData: [ //list of FileObject   
     
   ],  
   noOfSeen: <no of seen receiver>,  
   noOfDeliveries: <no of deliveried receiver>,  
   deliveryStatus: 0(sent)/ 1(delivered)/2(seen)  
   }  
   ],  
   totalNoOfReceipients: <total no of receipients of the chat>,  
   receipients: [ //list of UserObject  
     
   ]  
   }  
   onError🡪error object**
* **for a new chat request:  
  url: /messenger/chatrequest  
  method: POST  
  header: {  
   Authorization: <jwt token>,  
   u\_id : <user id>  
  }  
  query: none  
  body: {  
   receipients: [ //list of userIds  
   ]  
  }  
  response🡪  
  status:  
   OK 🡪 successful response  
   BAD\_REQUEST 🡪 target userId not found  
   UNAUTHORIZED 🡪 not all mutual following  
   INTERNAL\_SERVER\_ERROR 🡪 server error  
  body:   
  onSuccess:   
  format is same as response format of */messenger/chat* only deference is message list will be null  
  onError: ErrorObject**
* **for existing chat info updates:  
  url: /messenger/chatupdate  
  method: GET  
  headers: {  
   Authorization: <jwt token>,  
   u\_id : <user id>  
  }  
  query: none  
  body: {  
   chats: [ //list of chatids   
     
   ],  
   timestamp: <timestamp of the latest msg in the latest chat>  
  }  
  response🡪  
  status:  
   OK 🡪 successful response  
   BAD\_REQUEST 🡪 target chatid not found  
   UNAUTHORIZED 🡪 not all receipient  
   INTERNAL\_SERVER\_ERROR 🡪 server error  
  body:  
  onSuccess:  
  {  
   chats: [ // a list of chats format is same as */messenger/chat*  
     
   ]  
  }  
  onError: ErrorObject**
* **for requesting previous messages of chat:  
  url:/messenger/chat/message  
  method: GET  
  headers: {  
   Authorization: <jwt token>,  
   u\_id : <user id>  
  }  
  query: {  
   Id : <id of the requested chat>,  
   index: <starting index of message>   
  }  
  body: none  
  response🡪  
  status:  
   OK 🡪 successful response  
   BAD\_REQUEST 🡪 target chatid not found  
   UNAUTHORIZED 🡪 not receipient  
   INTERNAL\_SERVER\_ERROR 🡪 server error  
  body:  
  onSuccess:  
  {  
   index: <starting index of the msg list according to the database>,  
   length: <msg list length>,  
   id: <chat id>  
   msgs: [ //list of MessageObject  
     
   ]  
  }  
  onError: ErrorObject**
* **for front page request:**[**url:/messenger**](url:/messenger) **method: GET  
  header:  
  headers: {  
   Authorization: <jwt token>,  
   u\_id : <user id>  
  }  
  query: none  
  body: none  
  response🡪  
  status:  
   OK 🡪 successful response  
   BAD\_REQUEST 🡪 target id not found  
   UNAUTHORIZED 🡪 bad token  
   INTERNAL\_SERVER\_ERROR 🡪 server error  
  body:  
  onSuccess:  
  {  
   ref: <UserObject of the current user>,  
   noOfChats: <no of chats>,  
   chats: [ //list of ChatObject format same as */messenger/chat*  
     
   ]  
  }**