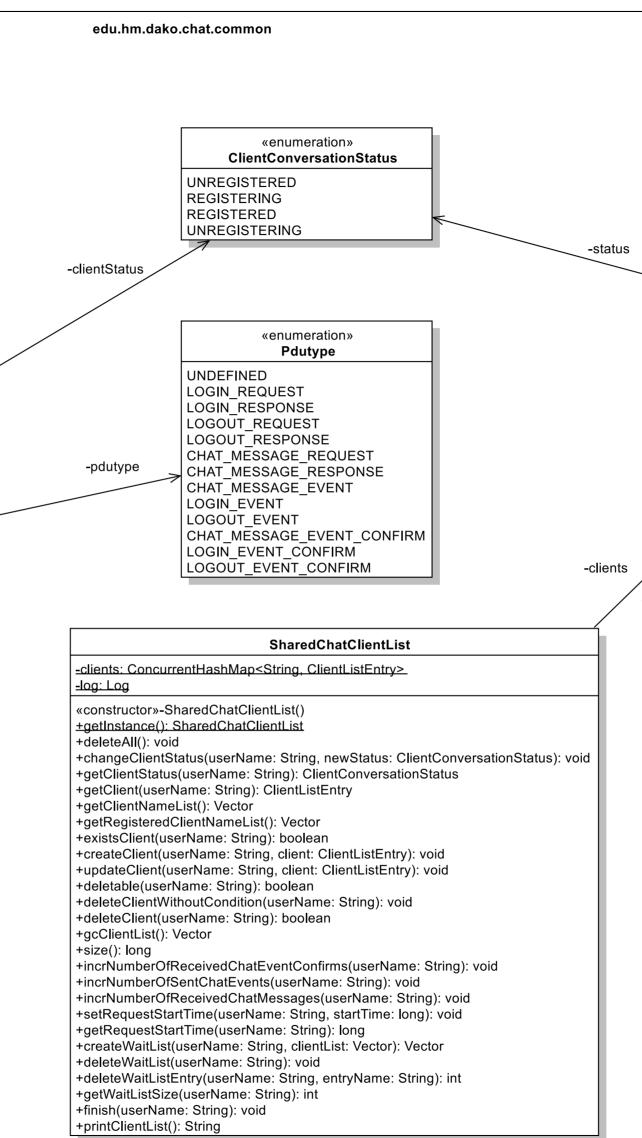**edu.hm.dako.chat.common**

ChatPDU

-serialVersionUID: long {readOnly}
-userName: String
-eventUserName: String
-clientThreadName: String
-serverThreadName: String
-sequenceNumber: long
-message: String
-clients: Vector
-serverTime: long
-errorCode: int
+NO_ERROR: int = 0 {readOnly}
+LOGIN_ERROR: int = 1 {readOnly}
-numberOfReceivedChatMessages: long
-numberOfSentEvents: long
-numberOfReceivedConfirms: long
-numberOfLostConfirms: long
-numberOfRetries: long

«constructor»+ChatPDU()
«constructor»+ChatPDU(cmd: Pdutype, clients: Vector)
«constructor»+ChatPDU(cmd: Pdutype, message: String)
+toString(): String
+printPdu(pdu: ChatPDU): void
+setClients(clients: Vector): void
+setPduType(pduType: Pdutype): void
+setUserName(userName: String): void
+setEventUserName(name: String): void
+setClientThreadName(threadName: String): void
+setServerThreadName(threadName: String): void
+setMessage(msg: String): void
+setServerTime(time: long): void
+setSequenceNumber(sequenceNumber: long): void
+getPduType(): Pdutype
+getClients(): Vector
+getUserName(): String
+getEventUserName(): String
+getClientThreadName(): String
+getServerThreadName(): String
+getMessage(): String
+getServerTime(): long
+getSequenceNumber(): long
+getClientStatus(): ClientConversationStatus
+setClientStatus(clientStatus: ClientConversationStatus): void
+getNumberOfSentEvents(): long
+setNumberOfSentEvents(nr: long): void
+getNumberOfReceivedConfirms(): long
+setNumberOfReceivedEventConfirms(nr: long): void
+getNumberOfLostConfirms(): long
+setNumberOfLostEventConfirms(nr: long): void
+getNumberOfRetries(): long
+setNumberOfRetries(nr: long): void
+getNumberOfReceivedChatMessages(): long
+setNumberOfReceivedChatMessages(nr: long): void
+getErrorCode(): int
+setErrorCode(code: int): void
+createLogoutEventPdu(userName: String, clientList: Vector, receivedPdu: ChatPDU): ChatPDU
+createLoginEventPdu(userName: String, clientList: Vector, receivedPdu: ChatPDU): ChatPDU
+createLoginResponsePdu(eventInitiator: String, receivedPdu: ChatPDU, clientList: Vector): ChatPDU
+createChatMessageEventPdu(userName: String, receivedPdu: ChatPDU): ChatPDU
+createLogoutResponsePdu(eventInitiator: String): ChatPDU
+createChatMessageResponsePdu(eventInitiator: String, clientThreadName: String, serverTime: long): ChatPDU
+createLoginErrorResponsePdu(receivedPdu: ChatPDU, errorCode: int): ChatPDU
+createLoginEventConfirm(userName: String, receivedPdu: ChatPDU): ChatPDU
+createLogoutEventConfirm(userName: String, receivedPdu: ChatPDU): ChatPDU
+createChatMessageEventConfirm(userName: String, receivedPdu: ChatPDU): ChatPDU

«enumeration» ClientConversationStatus

UNREGISTERED
REGISTERING
REGISTERED
UNREGISTERING

«enumeration» Pdutype

UNDEFINED
LOGIN_REQUEST
LOGIN_RESPONSE
LOGOUT_REQUEST
LOGOUT_RESPONSE
CHAT_MESSAGE_REQUEST
CHAT_MESSAGE_RESPONSE
CHAT_MESSAGE_EVENT
LOGIN_EVENT
LOGOUT_EVENT
CHAT_MESSAGE_EVENT_CONFIRM
LOGIN_EVENT_CONFIRM
LOGOUT_EVENT_CONFIRM

ClientListEntry

-userName: String
-session: Session
~finished: boolean
-loginTime: long
-startTime: long
-numberOfReceivedChatMessages: long
-numberOfSentEvents: long
-numberOfReceivedEventConfirms: long
-waitList: Vector

«constructor»+ClientListEntry(userName: String, session: Session)
+toString(): String
+setUserName(userName: String): void
+getUserName(): String
+setSession(session: Session): void
+getSession(): Session
+setLoginTime(time: long): void
+setStartTime(startTime: long): void
+getLoginTime(): long
+getStartTime(): long
+setNumberOfReceivedChatMessages(nr: long): void
+getNumberOfReceivedChatMessages(): long
+setNumberOfSentEvents(nr: long): void
+getNumberOfSentEvents(): long
+setNumberOfReceivedEventConfirms(nr: long): void
+getNumberOfReceivedEventConfirms(): long
+getStatus(): ClientConversationStatus
+setStatus(status: ClientConversationStatus): void
+isFinished(): boolean
+setFinished(finished: boolean): void
+incrNumberOfSentEvents(): void
+incrNumberOfReceivedEventConfirms(): void
+incrNumberOfReceivedChatMessages(): void
+setWaitList(list: Vector): void
+addWaitListEntry(userName: String): void
+getWaitList(): Vector
+clearWaitList(): void

SharedChatClientList

-clients: ConcurrentHashMap<String, ClientListEntry>
-log: Log

«constructor»-SharedChatClientList()
+getInstance(): SharedChatClientList
+deleteAll(): void
+changeClientStatus(userName: String, newStatus: ClientConversationStatus): void
+getClientStatus(userName: String): ClientConversationStatus
+getClient(userName: String): ClientListEntry
+getClientNameList(): Vector
+getRegisteredClientNameList(): Vector
+existsClient(userName: String): boolean
+createClient(userName: String, client: ClientListEntry): void
+updateClient(userName: String, client: ClientListEntry): void
+deletable(userName: String): boolean
+deleteClientWithoutCondition(userName: String): void
+deleteClient(userName: String): boolean
+gcClientList(): Vector
+size(): long
+incrNumberOfReceivedChatEventConfirms(userName: String): void
+incrNumberOfSentChatEvents(userName: String): void
+incrNumberOfReceivedChatMessages(userName: String): void
+setRequestStartTime(userName: String, startTime: long): void
+getRequestStartTime(userName: String): long
+createWaitList(userName: String, clientList: Vector): Vector
+deleteWaitList(userName: String): void
+deleteWaitListEntry(userName: String, entryName: String): int
+getWaitListSize(userName: String): int
+finish(userName: String): void
+printClientList(): String

-clientStatus
-pdutype
-status
-clients
1..*