edu.hm.dako.chat.benchmark

BenchmarkingClientImplAdvanced

-log: Log

#messageLength: int

#numberOfMessagesToSend: int

#clientThinkTime: int #resultPath: String

-chatResponseReceived: AtomicBoolean

-chatPduRtt: long[*]

«constructor»+BenchmarkingClientImplAdvanced(uri: URI, messageLength: int, numberOfMessages: int, clientThinkTime: int, numberOfClient: int, resultPath: String)

+onOpen(): void

+onError(t: Throwable): void

+onClose(): void

+onMessage(receivedPdu: ChatPDU): void

+executeTest(): void

-printStatistic(): void

-waitUntilChatResponseReceived(): void

-sendMessageAndWaitForAck(i: int): void

#chatMessageResponseAction(receivedPdu: ChatPDU): void

#chatMessageEventAction(receivedPdu: ChatPDU): void

-confirmChatMessageEvent(receivedPdu: ChatPDÚ): void

-confirmLoginEvent(receivedPdu: ChatPDU): void

-confirmLogoutEvent(receivedPdu: ChatPDÚ): void #loginResponseAction(receivedPdu: ChatPDU): void

#loginEventAction(receivedPdu: ChatPDU): void

#logoutEventAction(receivedPdu: ChatPDU): void #logoutResponseAction(receivedPdu: ChatPDU): void

+handleIncomingPdu(receivedPdu: ChatPDU): void

BenchmarkingClientImplSimple

-log: Log

#messageLength: int

#numberOfMessagesToSend: int

#clientThinkTime: int

#resultPath: String

-chatResponseReceived: AtomicBoolean

-chatPduRtt: long[*]

«constructor»+BenchmarkingClientImplSimple(uri: URI, messageLength: int, numberOfMessages: int, clientThinkTime: int, numberOfClient: int, resultPath: String)

+onOpen(session: Session): void

+onError(t: Throwable): void

+onClose(): void

+onMessage(receivedPdu: ChatPDU): void

+executeTest(): void

-printStatistic(): void

-waitUntilChatResponseReceived(): void

-sendMessageAndWaitForAck(i: int): void

#chatMessageResponseAction(receivedPdu: ChatPDU): void

#chatMessageEventAction(receivedPdu: ChatPDU): void #loginResponseAction(receivedPdu: ChatPDU): void

#loginEventAction(receivedPdu: ChatPDU): void

#logoutEventAction(receivedPdu: ChatPDU): void

#logoutResponseAction(receivedPdu: ChatPDU): void

+handleIncomingPdu(receivedPdu: ChatPDU): void

-log: Log

#messageCounter: AtomicInteger

#loginEvents: AtomicInteger

#serverPort: int

#remoteServerAddress: String

+login(name: String): void

+logout(name: String): void

+tell(name: String, text: String): void

+isLoggedOut(): boolean

#sendPduToServer(client: String, pdu: ChatPDU): void

#chatMessageResponseAction(receivedPdu: ChatPDU): void #chatMessageEventAction(receivedPdu: ChatPDU): void

#loginEventAction(receivedPdu: ChatPDU): void

#logoutResponseAction(receivedPdu: ChatPDU): void

AbstractChatClient

#status: ClientConversationStatus

#userName: String #session: Session

#threadName: String

#eventCounter: AtomicInteger

#messageEvents: AtomicInteger

#confirmCounter: AtomicInteger

#logoutEvents: AtomicInteger

#localPort: int

#finished: boolean = false

+executeTest(): void

#loginResponseAction(receivedPdu: ChatPDU): void

#logoutEventAction(receivedPdu: ChatPDU): void

Result

-userName: String

-implementation: String -totalEvents: int

-sendMessages: int

-loginEvents: int -logoutEvents: int -chatPduRtt: long[*]

«constructor»+Result()

+aetImplementation(): String

+setImplementation(implementation: String): void

+getChatPduRtt(): long[*]

+setChatPduRtt(chatPduRtt: long[*]): void

+getUserName(): String

+setUserName(userName: String): void

+getTotalEvents(): int

+setTotalEvents(totalEvents: int): void

+getSendMessages(): int

+setSendMessages(sendMessages: int): void

+getLoginEvents(): int

ClientCommunication

+setLoginEvents(loginEvents: int): void

+getLogoutEvents(): int

+setLogoutEvents(logoutEvents: int): void