When you first run your programs, please print either “house” or “user;<username>”

Messages

- Auction House -> Bank
    - GET_ITEMS_FROM_BANK houseId itemCount
        - Description: Returns items from the bank
        - houseId: (int) The id of the house requesting this information
        - itemCount: (int) The number of items desired
        - Invoked by: self
        - Expected Back: A command ITEMS with a list of items
    - VALID_BID houseId userId itemBid itemId previousHighBidUserId previousBid
        - Description: Message to the Bank indicating a bid from the user is a valid bid and should be executed if possible.
        - houseId: (int) The id of the house requesting this information
        - userId: (int) The id of the bidding user
        - itemBid: (double) The bid being made on the said item
        - itemId: (int) The id of the desired item
        - previousHighBidUserId: (int) The id of the current highest bidder
        - previousBid: (double) The bid currently on the item
        - Invoked by: ‘BID …’ from user
        - Expected back: ‘ACCEPT_BID …’ or ‘REJECT_BID …’ from bank
    - AUCTION_ENDED houseId winnerId itemId itemBid itemName
        - Description: Message sent to the bank if the auctioning of a certain item has ended because of the 30 second wait time elapsing
        - houseId: (int) The id of the house
        - winnerId: (int) User id of the current highest bidder
        - itemId: (int) The id of the item
        - itemBid: (double) The current highest bid on the item
        - Invoked by: self
        - Expected back: nothing (Maybe REMOVE_ITEM)
    - EXIT houseID
        - Description: Message to the bank requesting to exit the network
        - houseId: (int) The id of the house
        - Invoked by: self
        - Expected back: ‘CAN_EXIT’ if the house can exit, ‘ERROR …’ otherwise from the bank
- Bank -> Auction House
    - ITEMS houseId itemCount <itemName itemId itemBid itemDesc>
        - Description: Sends the items currently on sale as a list
        - houseId: (int) The id of the house
        - itemCount: (int) The number of items
        - itemName: (String) The String name of the item
        - itemId: (int) The id of the item
        - itemBid: (double) The current highest bid on the item
        - bidUserId: (int) User id of the current highest bidder
        - itemDesc: (String) Description of the item
        - Invoked by: ‘GET_ITEMS’ from bank

        - Expected Back: nothing
    - ACCEPT_BID userId itemId itemBid
        - Description: Indicates that the bid made on this item was valid
        - userId: (int) The id of the bidding user
        - itemId: (int) The id of the desired item
        - itemBid: (double) The bid being made on that item
        - Invoked by: ‘VALID_BID …’ to bank
        - Expected back: nothing
    - REJECT_BID userId itemId
        - Description: Indicates that the bid made on this item was invalid
        - userId: (int) The id of the bidding user
        - itemId: (int) The id of the desired item
        - Invoked by: ‘VALID_BID …’ to bank
        - Expected back: nothing
    - LOGIN id
        - Description: Sends a generated id (stored as the houseId)
        - id: (int) What will become the houseId
        - Invoked by: “house” message on start-up of socket
        - Expected back: nothing
    - CAN_EXIT
        - Description: Message to house that it can exit the program
        - Invoked by: ‘EXIT houseID’ by house
        - Expected back: nothing
    - ERROR message
        - Description: Something went wrong message
        - message: (String) Error message
        - Invoked by: ??
        - Expected back: ??
- Bank -> User
    - Missing the command to return the list of houses…(HOUSE_LIST)
    - HOUSE_LIST <houseId hostName port>
        - Description: Returns a list of all the relevant information for all the houses
        - houseId: (int) id of the house with the item
        - hostName: (String) The hostname of the auction house
        - port: (int) The port that the auction house is open on
        - Invoked by: ‘GET_HOUSES …’ by user
        - Expected back: nothing
    - OUTBID houseId itemId outBidderName amount lastHighBidderId
        - Description: A message stating that for a certain item on a certain house, the last bidder was outbid
            - This will be sent to everyone so they know who the top bidder is and the top amount. This is irrelevant of who the user is (not only sent to the user outbid)
        - houseId: (int) id of the house with the item
        - itemId: (int) id of the item that was bid on
        - outBidderId: (int) id of the new highest bidder on the specified item
        - amount: (double) The new highest bid for the item

        - Invoked by: ‘BID …’ from any user
        - Expected back: nothing
    - WINNER houseId itemName itemId amount
        - Description: A message stating, to the person who won the item, that they won the item
        - houseId: (int) id of the house with the item
        - itemName: (String) The name of the item being bit
        - itemId: (int) The id of the item won
        - amount: (double) The amount that the item was sold for
        - Invoked by: ‘AUCTION_WON …’ by house
        - Expected back: nothing
    - ITEM_WON houseId itemName itemId winnerUsername amount
        - Description: A message stating that a certain item was won and is no longer auctionable
        - houseId: (int) id of the house with the item
        - itemName: (String) The name of the item being bit
        - itemId: (int) The id of the item won
        - winnerUsername: (String) The username of the winning bidder
        - amount: (double) The amount that the item was sold for
        - Invoked by: ‘AUCTION_WON …’ by house
        - Expected back: nothing
    - LOGIN id
        - Description: Sends a generated id (stored as the userId)
        - id: (int) What will become the userId
        - Invoked by: “user” message on start-up of socket
        - Expected back: nothing
    - CAN_EXIT
        - Description: Message to user that it can exit the program
        - Invoked by: ‘EXIT userID’ by user
        - Expected back: nothing
    - ERROR message
        - Description: Something went wrong
        - message: The error message
        - Invoked by: ??
        - Expected back: ??
- User -> Bank
    - GET_HOUSES userId
        - Description: Request for the list of all of the houses by id
        - userId: (int) The id of the requesting user
        - Invoked by: self
        - Expected back: ‘HOUSE_LIST …’
    - EXIT userID
        - Description: Message to the bank requesting to exit the network
        - userId: (int) The id of the user
        - Invoked by: self
        - Expected back: ‘CAN_EXIT’ if the user can exit, ‘ERROR …’ otherwise from the bank

- User -> House
    - BID userId bidAmount itemId houseId
        - Description: A request by the user to bid on an item
        - userId: (int) The id of the user
        - bidAmount: (double) the requested bid amount
        - itemId: (int) The id of the item
        - houseId: (int) The id of the house with the specified item
        - Invoked by: self
        - Expected back: ‘ACCEPT …’ or ‘REJECT …’ by bank
    - GET_ITEMS
        - Description: Asks an auction house for the items that it currently is selling
        - Invoked by: ‘GET_ITEMS userId houseId’ by user
        - Expected back: ‘ITEMS …’
    - GET_ITEM itemID
        - Description: Asks an auction house for a specific item and its information
        - itemId: (int) The id of the requested item
        - Invoked by: ‘BID ...’ by user
        - Expected back: ‘ITEM …’
    - EXIT userID
        - Description: Message to the bank requesting to exit the network
        - userId: (int) The id of the user
        - Invoked by: self
        - Expected back: ‘CAN_EXIT’ if the user can exit, ‘ERROR …’ otherwise from the bank
- House -> User (UPDATE: added itemCount to ITEMS)
    - ITEMS houseId itemCount <itemName itemId itemBid bidUserId itemDesc>
        - Description: Sends the items currently on sale as a list
        - houseId: (int) The id of the house
        - itemCount: (int) The number of items
        - itemName: (String) The String name of the item
        - itemId: (int) The id of the item
        - itemBid: (double) The current highest bid on the item
        - bidUserId: (int) User id of the current highest bidder
        - itemDesc: (String) Description of the item
        - Invoked by: ‘GET_ITEMS’ from bank
        - Expected Back: nothing
    - ACCEPT houseId itemId
        - Description: A message stating the bid was valid (not rejected)
        - houseId: (int) id of the house with the item
        - itemId: (int) id of the item that was bid on
        - Invoked by: ‘BID ...’ by user -> ‘ITEM ...’ by house
        - Expected back: nothing
    - REJECT houseId itemId
        - Description: A message stating the bid was rejected (for whatever reason)
        - houseId: (int) id of the house with the item
        - itemId: (int) id of the item that was bid on
        - Invoked by: ‘BID ...’ by user -> ‘ITEM ...’ by house

        - Expected back: nothing
    - ITEM houseId itemName itemID itemBid itemBidUser itemDesc
        - Description: Sends the details of a specific item
        - houseId: (int) The id of the house
        - itemName: (String) The String name of the item
        - itemId: (int) The id of the item
        - itemBid: (double) The current highest bid on the item
        - bidUserId: (int) User id of the current highest bidder
        - itemDesc: (String) Description of the item
        - Invoked by: ‘GET_ITEM itemId’ from bank
        - Expected back: nothing
    - CAN_EXIT houseId
        - Description: Message to user that it can exit the program
        - houseId: (int) id of the calling house
        - Invoked by: ‘EXIT userID’ by user
        - Expected back: nothing

When getting a string…

reader.getLine.split(“;”)

Delimiter: {“;”}