

Bank
resourceDeck : HashMap<ResourceType, Integer> developmentCards : ArrayList<DevelopmentCards>
canBuyDevelopmentCard()

Hex
resource : ResourceType location : Point number : int

Game
playerList : ArrayList<Player> theMap : Map bank : Bank currentTurnIndex : int turnTracker: TurnTracker log: MessageList chat: MessageList winner : String
canCreateUser() canAuthenticateUser()

Player
name : String color : CatanColor playerIndex : int developmentCardPlayed : boolean victoryPoints : int resourceHand : HashMap<ResourceType, Integer> playableDevCardHand : HashMap<DevCardType, Integer> unplayableDevCardHand : HashMap<DevCardType, Integer> piecesAvailable : HashMap<PieceType, Integer>
canPlaceCity() canPlaceSettlement() canPlaceRoad() canBuyDevelopmentCard canTradeWithPlayer(playerId : int) canTradeWithBank() canPlayDevelopmentCard(type : DevelopmentCardType) canRollDice() canSendMessage(message : String) canEndTurn()

ModelUpdater
updateModel(json: json) updateBank(json: json) updatePlayer(json: json) updateGame(json: json) updateMap(json: json)

MessageList
messages:ArrayList<String>

Port
ratio : int resource : ResourceType direction : VertexDirection location : HexLocation

Robber
location : HexLocation

Map
hexList : ArrayList<Hex> radius : int portList : List<Port>
canGetRolledResources(diceRoll : int, playerId : int)

Trade Offer
sender : int receiver : int offer : HashMap<ResourceType, Integer>
int getSender() void setSender(int sender) int getReceiver() void setReceiver(int receiver) HashMap<ResourceType, Integer> getOffer() void setOffer(HashMap<ResourceType, Integer> offer)

City
location: VertexLocation

Settlement
location: VertexLocation

Road
location: EdgeLocation

DevCard
type: DevCardType holder: Player

Turn Tracker
status: GameStatus currentTurn: int longestRoad: DevCard largestArmy;; DevCard

InvalidOptionException