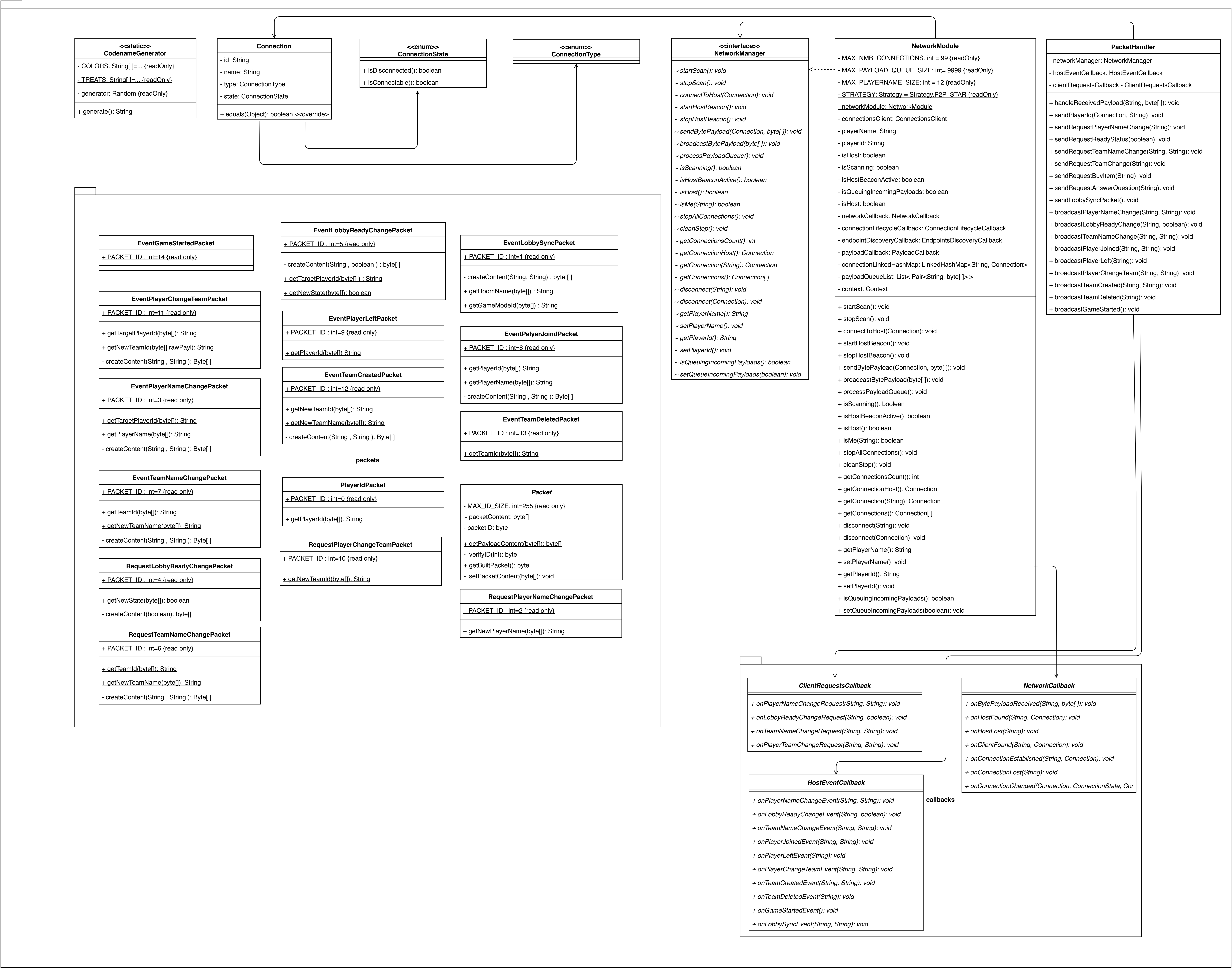

<<static>>
CodenameGenerator
- COLORS: String[ ]=... {readOnly}
- TREATS: String[ ]=... {readOnly}
- generator: Random {readOnly}
+ generate(): String
Connection
- id: String
- name: String
- type: ConnectionType
- state: ConnectionState
+ equals(Object): boolean <<override>
<<enum>>
ConnectionState
+ isDisconnected(): boolean
+ isConnectable(): boolean
<<enum>>
ConnectionType
<<interface>>
NetworkManager
~ startScan(): void
~ stopScan(): void
~ connectToHost(Connection): void
~ startHostBeacon(): void
~ stopHostBeacon(): void
~ sendBytePayload(Connection, byte[ ]): void
~ broadcastBytePayload(byte[ ]): void
~ processPayloadQueue(): void
~ isScanning(): boolean
~ isHostBeaconActive(): boolean
~ isHost(): boolean
~ isMe(String): boolean
~ stopAllConnections(): void
~ cleanStop(): void
~ getConnectionsCount(): int
~ getConnectionHost(): Connection
~ getConnection(String): Connection
~ getConnections(): Connection[ ]
~ disconnect(String): void
~ disconnect(Connection): void
~ getPlayerName(): String
~ setPlayerName(): void
~ getPlayerId(): String
~ setPlayerId(): void
~ isQueuingIncomingPayloads(): boolean
~ setQueueIncomingPayloads(boolean): void
NetworkModule
- MAX_NMB_CONNECTIONS: int = 99 {readOnly}
- MAX_PAYLOAD_QUEUE_SIZE: int= 9999 {readOnly}
- MAX_PLAYERNAME_SIZE: int = 12 {readOnly}
- STRATEGY: Strategy = Strategy.P2P_STAR {readOnly}
- networkModule: NetworkModule
- connectionsClient: ConnectionsClient
- playerName: String
- playerId: String
- isHost: boolean
- isScanning: boolean
- isHostBeaconActive: boolean
- isQueuingIncomingPayloads: boolean
- isHost: boolean
- networkCallback: NetworkCallback
- connectionLifecycleCallback: ConnectionLifecycleCallback
- endpointDiscoveryCallback: EndpointsDiscoveryCallback
- payloadCallback: PayloadCallback
- connectionLinkedHashMap: LinkedHashMap<String, Connection>
- payloadQueueList: List< Pair<String, byte[ ]> >
- context: Context
+ startScan(): void
+ stopScan(): void
+ connectToHost(Connection): void
+ startHostBeacon(): void
+ stopHostBeacon(): void
+ sendBytePayload(Connection, byte[ ]): void
+ broadcastBytePayload(byte[ ]): void
+ processPayloadQueue(): void
+ isScanning(): boolean
+ isHostBeaconActive(): boolean
+ isHost(): boolean
+ isMe(String): boolean
+ stopAllConnections(): void
+ cleanStop(): void
+ getConnectionsCount(): int
+ getConnectionHost(): Connection
+ getConnection(String): Connection
+ getConnections(): Connection[ ]
+ disconnect(String): void
+ disconnect(Connection): void
+ getPlayerName(): String
+ setPlayerName(): void
+ getPlayerId(): String
+ setPlayerId(): void
+ isQueuingIncomingPayloads(): boolean
+ setQueueIncomingPayloads(boolean): void
PacketHandler
- networkManager: NetworkManager
- hostEventCallback: HostEventCallback
- clientRequestsCallback - ClientRequestsCallback
+ handleReceivedPayload(String, byte[ ]): void
+ sendPlayerId(Connection, String): void
+ sendRequestPlayerNameChange(String): void
+ sendRequestReadyStatus(boolean): void
+ sendRequestTeamNameChange(String, String): void
+ sendRequestTeamChange(String): void
+ sendRequestBuyItem(String): void
+ sendRequestAnswerQuestion(String): void
+ sendLobbySyncPacket(): void
+ broadcastPlayerNameChange(String, String): void
+ broadcastLobbyReadyChange(String, boolean): void
+ broadcastTeamNameChange(String, String): void
+ broadcastPlayerJoined(String, String): void
+ broadcastPlayerLeft(String): void
+ broadcastPlayerChangeTeam(String, String): void
+ broadcastTeamCreated(String, String): void
+ broadcastTeamDeleted(String): void
+ broadcastGameStarted(): void
EventGameStartedPacket
+ PACKET_ID : int=14 {read only}
EventLobbyReadyChangePacket
+ PACKET_ID : int=5 {read only}
- createContent(String , boolean) : byte[ ]
+ getTargetPlayerId(byte[]) : String
+ getNewState(byte[]): boolean
EventLobbySyncPacket
+ PACKET_ID : int=1 {read only}
- createContent(String, String) : byte [ ]
+ getRoomName(byte[]) : String
+ getGameModeId(byte[]) : String
EventPlayerChangeTeamPacket
+ PACKET_ID : int=11 {read only}
+ getTargetPlayerId(byte[]): String
+ getNewTeamId(byte[] rawPayl): String
- createContent(String , String): Byte[ ]
EventPlayerLeftPacket
+ PACKET_ID : int=9 {read only}
+ getPlayerId(byte[]) String
EventPalyerJoindPacket
+ PACKET_ID : int=8 {read only}
+ getPlayerId(byte[]) String
+ getPlayerName(byte[]): String
- createContent(String , String): Byte[ ]
EventPlayerNameChangePacket
+ PACKET_ID : int=3 {read only}
+ getTargetPlayerId(byte[]): String
+ getPlayerName(byte[]): String
- createContent(String , String): Byte[ ]
EventTeamCreatedPacket
+ PACKET_ID : int=12 {read only}
+ getNewTeamId(byte[]): String
+ getNewTeamName(byte[]): String
- createContent(String , String): Byte[ ]
EventTeamDeletedPacket
+ PACKET_ID : int=13 {read only}
+ getTeamId(byte[]): String
packets
EventTeamNameChangePacket
+ PACKET_ID : int=7 {read only}
+ getTeamId(byte[]): String
+ getNewTeamName(byte[]): String
- createContent(String , String): Byte[ ]
PlayerIdPacket
+ PACKET_ID : int=0 {read only}
+ getPlayerId(byte[]): String
Packet
- MAX_ID_SIZE: int=255 {read only}
~ packetContent: byte[]
- packetID: byte
+ getPayloadContent(byte[]): byte[]
- verifyID(int): byte
+ getBuiltPacket(): byte
~ setPacketContent(byte[]): void
RequestPlayerChangeTeamPacket
+ PACKET_ID : int=10 {read only}
+ getNewTeamId(byte[]): String
RequestLobbyReadyChangePacket
+ PACKET_ID : int=4 {read only}
+ getNewState(byte[]): boolean
- createContent(boolean): byte[]
RequestPlayerNameChangePacket
+ PACKET_ID : int=2 {read only}
+ getNewPlayerName(byte[]): String
RequestTeamNameChangePacket
+ PACKET_ID : int=6 {read only}
+ getTeamId(byte[]): String
+ getNewTeamName(byte[]): String
- createContent(String , String): Byte[ ]
ClientRequestsCallback
+ onPlayerNameChangeRequest(String, String): void
+ onLobbyReadyChangeRequest(String, boolean): void
+ onTeamNameChangeRequest(String, String): void
+ onPlayerTeamChangeRequest(String, String): void
NetworkCallback
+ onBytePayloadReceived(String, byte[ ]): void
+ onHostFound(String, Connection): void
+ onHostLost(String): void
+ onClientFound(String, Connection): void
+ onConnectionEstablished(String, Connection): void
+ onConnectionLost(String): void
+ onConnectionChanged(Connection, ConnectionState, Cor
HostEventCallback
+ onPlayerNameChangeEvent(String, String): void
+ onLobbyReadyChangeEvent(String, boolean): void
+ onTeamNameChangeEvent(String, String): void
+ onPlayerJoinedEvent(String, String): void
+ onPlayerLeftEvent(String): void
+ onPlayerChangeTeamEvent(String, String): void
+ onTeamCreatedEvent(String, String): void
+ onTeamDeletedEvent(String): void
+ onGameStartedEvent(): void
+ onLobbySyncEvent(String, String): void
callbacks