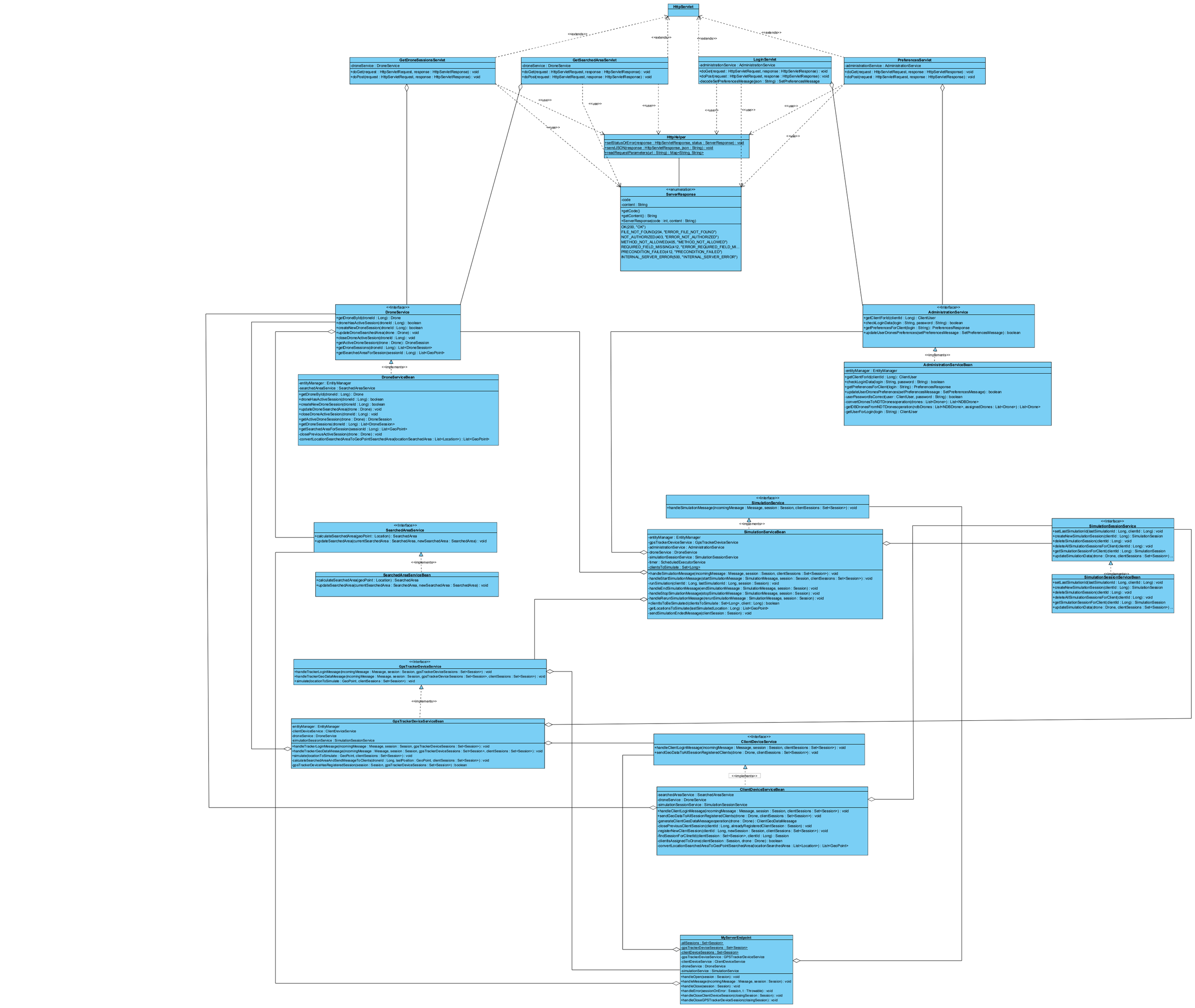
HttpServlet
<<extends>>
GetDroneSessionsServlet
-droneService : DroneService
+doGet(request : HttpServletRequest, response : HttpServletResponse) : void
+doPost(request : HttpServletRequest, response : HttpServletResponse) : void
GetSearchedAreaServlet
-droneService : DroneService
+doGet(request : HttpServletRequest, response : HttpServletResponse) : void
+doPost(request : HttpServletRequest, response : HttpServletResponse) : void
LoginServlet
-administrationService : AdministrationService
+doGet(request : HttpServletRequest, response : HttpServletResponse) : void
+doPost(request : HttpServletRequest, response : HttpServletResponse) : void
-decodeSetPreferencesMessage(json : String) : SetPreferencesMessage
PreferencesServlet
-administrationService : AdministrationService
+doGet(request : HttpServletRequest, response : HttpServletResponse) : void
+doPost(request : HttpServletRequest, response : HttpServletResponse) : void
<<use>>
HttpHelper
+setStatusOfError(response : HttpServletResponse, status : ServerResponse) : void
+sendJSON(response : HttpServletResponse, json : String) : void
+readRequestParameters(url : String) : Map<String, String>
<<enumeration>>
ServerResponse
-code
-content : String
+getCode() : int
+getContent() : String
+ServerResponse(code : int, content : String)
OK(200, "OK")
FILE_NOT_FOUND(204, "ERROR_FILE_NOT_FOUND")
NOT_AUTHORIZED(403, "ERROR_NOT_AUTHORIZED")
METHOD_NOT_ALLOWED(405, "METHOD_NOT_ALLOWED")
REQUIRED_FIELD_MISSING(412, "ERROR_REQUIRED_FIELD_MI...
PRECONDITION_FAILED(412, "PRECONDITION_FAILED")
INTERNAL_SERVER_ERROR(500, "INTERNAL_SERVER_ERROR")
<<interface>>
DroneService
+getDroneById(droneId : Long) : Drone
+droneHasActiveSession(droneId : Long) : boolean
+createNewDroneSession(droneId : Long) : boolean
+updateDroneSearchedArea(drone : Drone) : void
+closeDroneActiveSession(droneId : Long) : void
+getActiveDroneSession(drone : Drone) : DroneSession
+getDroneSessions(droneId : Long) : List<DroneSession>
+getSearchedAreaForSession(sessionId : Long) : List<GeoPoint>
<<implements>>
DroneServiceBean
-entityManager : EntityManager
-searchedAreaService : SearchedAreaService
+getDroneById(droneId : Long) : Drone
+droneHasActiveSession(droneId : Long) : boolean
+createNewDroneSession(droneId : Long) : boolean
+updateDroneSearchedArea(drone : Drone) : void
+closeDroneActiveSession(droneId : Long) : void
+getActiveDroneSession(drone : Drone) : DroneSession
+getDroneSessions(droneId : Long) : List<DroneSession>
+getSearchedAreaForSession(sessionId : Long) : List<GeoPoint>
-closePreviousActiveSession(drone : Drone) : void
-convertLocationSearchedAreaToGeoPoint(searchedArea(locationSearchedArea : List<Location>) : List<GeoPoint>
<<interface>>
AdministrationService
+getClientForId(clientId : Long) : ClientUser
+checkLoginData(login : String, password : String) : boolean
+getPreferencesForClient(login : String) : PreferencesResponse
+updateUserDronesPreferences(setPreferencesMessage : SetPreferencesMessage) : boolean
AdministrationServiceBean
-entityManager : EntityManager
+getClientForId(clientId : Long) : ClientUser
+checkLoginData(login : String, password : String) : boolean
+getPreferencesForClient(login : String) : PreferencesResponse
+updateUserDronesPreferences(setPreferencesMessage : SetPreferencesMessage) : boolean
-userPasswordIsCorrect(user : ClientUser, password : String) : boolean
-convertDronesToNDTDronesoperation(drones : List<Drone>) : List<NDBDrone>
-getDIEDronesFromNDTDronesoperation(ndbDrones : List<NDBDrone>, assignedDrones : List<Drone>) : List<Drone>
-getUserForLogin(login : String) : ClientUser
<<interface>>
SimulationService
+handleSimulationMessage(incomingMessage : Message, session : Session, clientSessions : Set<Session>) : void
SimulationServiceBean
-entityManager : EntityManager
-gpsTrackerDeviceService : GpsTrackerDeviceService
-administrationService : AdministrationService
-droneService : DroneService
-simulationSessionService : SimulationSessionService
-timer : ScheduledExecutorService
-clientsToSimulate : Set<Long>
+handleSimulationMessage(incomingMessage : Message, session : Session, clientSessions : Set<Session>) : void
-handleStartSimulationMessage(startSimulationMessage : SimulationMessage, session : Session, clientSessions : Set<Session>) : void
-runSimulation(clientId : Long, lastSimulationId : Long, session : Session) : void
-handleEndSimulationMessage(endSimulationMessage : SimulationMessage, session : Session) : void
-handleStopSimulationMessage(stopSimulationMessage : SimulationMessage, session : Session) : void
-handleResumeSimulationMessage(resumeSimulationMessage : SimulationMessage, session : Session) : void
+clientsToBeSimulated(clientsToSimulate : Set<Long>, client : Long) : boolean
-getLocationsToSimulate(lastSimulatedLocation : Long) : List<GeoPoint>
-sendSimulationEndedMessage(clientSession : Session) : void
<<interface>>
SimulationSessionService
+setLastSimulationId(lastSimulationId : Long, clientId : Long) : void
+createNewSimulationSession(clientId : Long) : SimulationSession
+deleteSimulationSession(clientId : Long) : void
+deleteAllSimulationSessionsForClient(clientId : Long) : void
+getSimulationSessionForClient(clientId : Long) : SimulationSession
+updateSimulationData(drone : Drone, clientSessions : Set<Session>) ...
SimulationSessionServiceBean
+setLastSimulationId(lastSimulationId : Long, clientId : Long) : void
+createNewSimulationSession(clientId : Long) : SimulationSession
+deleteSimulationSession(clientId : Long) : void
+deleteAllSimulationSessionsForClient(clientId : Long) : void
+getSimulationSessionForClient(clientId : Long) : SimulationSession
+updateSimulationData(drone : Drone, clientSessions : Set<Session>) ...
<<interface>>
SearchedAreaService
+calculateSearchedArea(geoPoint : Location) : SearchedArea
+updateSearchedArea(currentSearchedArea : SearchedArea, newSearchedArea : SearchedArea) : void
SearchedAreaServiceBean
+calculateSearchedArea(geoPoint : Location) : SearchedArea
+updateSearchedArea(currentSearchedArea : SearchedArea, newSearchedArea : SearchedArea) : void
<<interface>>
GpsTrackerDeviceService
+handleTrackerLoginMessage(incomingMessage : Message, session : Session, gpsTrackerDeviceSessions : Set<Session>) : void
+handleTrackerGeoDataMessage(incomingMessage : Message, session : Session, gpsTrackerDeviceSessions : Set<Session>, clientSessions : Set<Session>) : void
+simulateLocationToSimulate(locationToSimulate : GeoPoint, clientSessions : Set<Session>) : void
GpsTrackerDeviceServiceBean
-entityManager : EntityManager
-clientDeviceService : ClientDeviceService
-droneService : DroneService
-simulationSessionService : SimulationSessionService
+handleTrackerLoginMessage(incomingMessage : Message, session : Session, gpsTrackerDeviceSessions : Set<Session>) : void
+handleTrackerGeoDataMessage(incomingMessage : Message, session : Session, gpsTrackerDeviceSessions : Set<Session>, clientSessions : Set<Session>) : void
+simulateLocationToSimulate(locationToSimulate : GeoPoint, clientSessions : Set<Session>) : void
-calculateSearchedAreaAndSendMessageToClients(droneId : Long, lastPosition : GeoPoint, clientSessions : Set<Session>) : void
-gpsTrackerDeviceHasRegisteredSession(session : Session, gpsTrackerDeviceSessions : Set<Session>) : boolean
<<interface>>
ClientDeviceService
+handleClientLoginMessage(incomingMessage : Message, session : Session, clientSessions : Set<Session>) : void
+sendGeoDataToAllSessionRegisteredClients(drone : Drone, clientSessions : Set<Session>) : void
ClientDeviceServiceBean
-searchedAreaService : SearchedAreaService
-droneService : DroneService
-simulationSessionService : SimulationSessionService
+handleClientLoginMessage(incomingMessage : Message, session : Session, clientSessions : Set<Session>) : void
+sendGeoDataToAllSessionRegisteredClients(drone : Drone, clientSessions : Set<Session>) : void
-generateClientGeoDataMessage(operation(drone : Drone) : ClientGeoDataMessage
-closePreviousClientSession(clientId : Long, alreadyRegisteredClientSession : Session) : void
-registerNewClientSession(clientId : Long, newSession : Session, clientSessions : Set<Session>) : void
-findSessionForClientId(clientSession : Set<Session>, clientId : Long) : Session
-clientsAssignedToDrone(clientSession : Session, drone : Drone) : boolean
-convertLocationSearchedAreaToGeoPoint(searchedArea(locationSearchedArea : List<Location>) : List<GeoPoint>
MyServerEndpoint
-allSessions : Set<Session>
-gpsTrackerDeviceSessions : Set<Session>
-clientDeviceSessions : Set<Session>
-gpsTrackerDeviceService : GPSTrackerDeviceService
-clientDeviceService : ClientDeviceService
-droneService : DroneService
-simulationService : SimulationService
+handleOpen(session : Session) : void
+handleMessage(incomingMessage : Message, session : Session) : void
+handleClose(session : Session) : void
+handleError(session, Throwable) : void
+handleCloseClientDeviceSession(closingSession : Session) : void
+handleCloseGPSTrackerDeviceSession(closingSession) : void