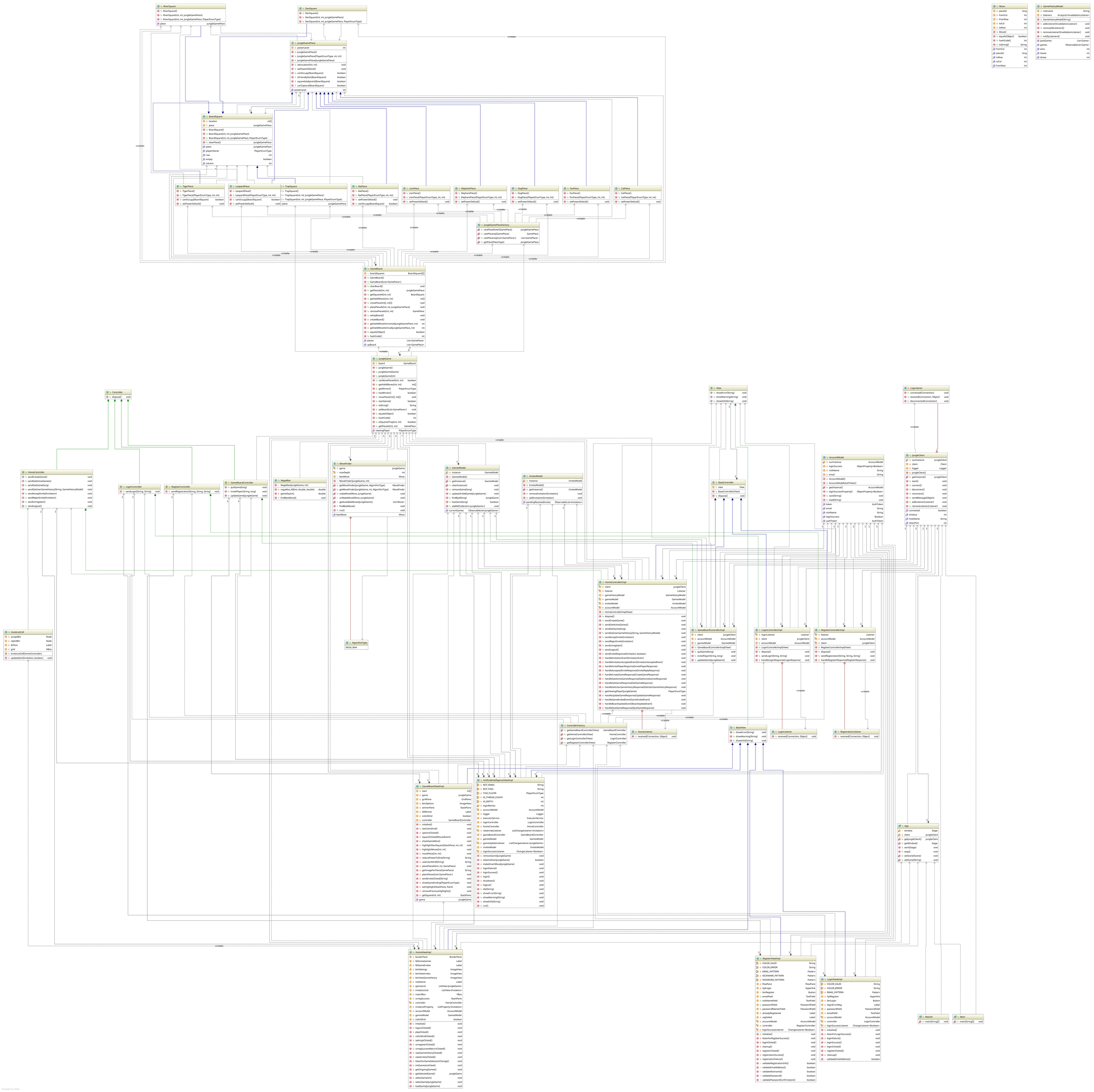
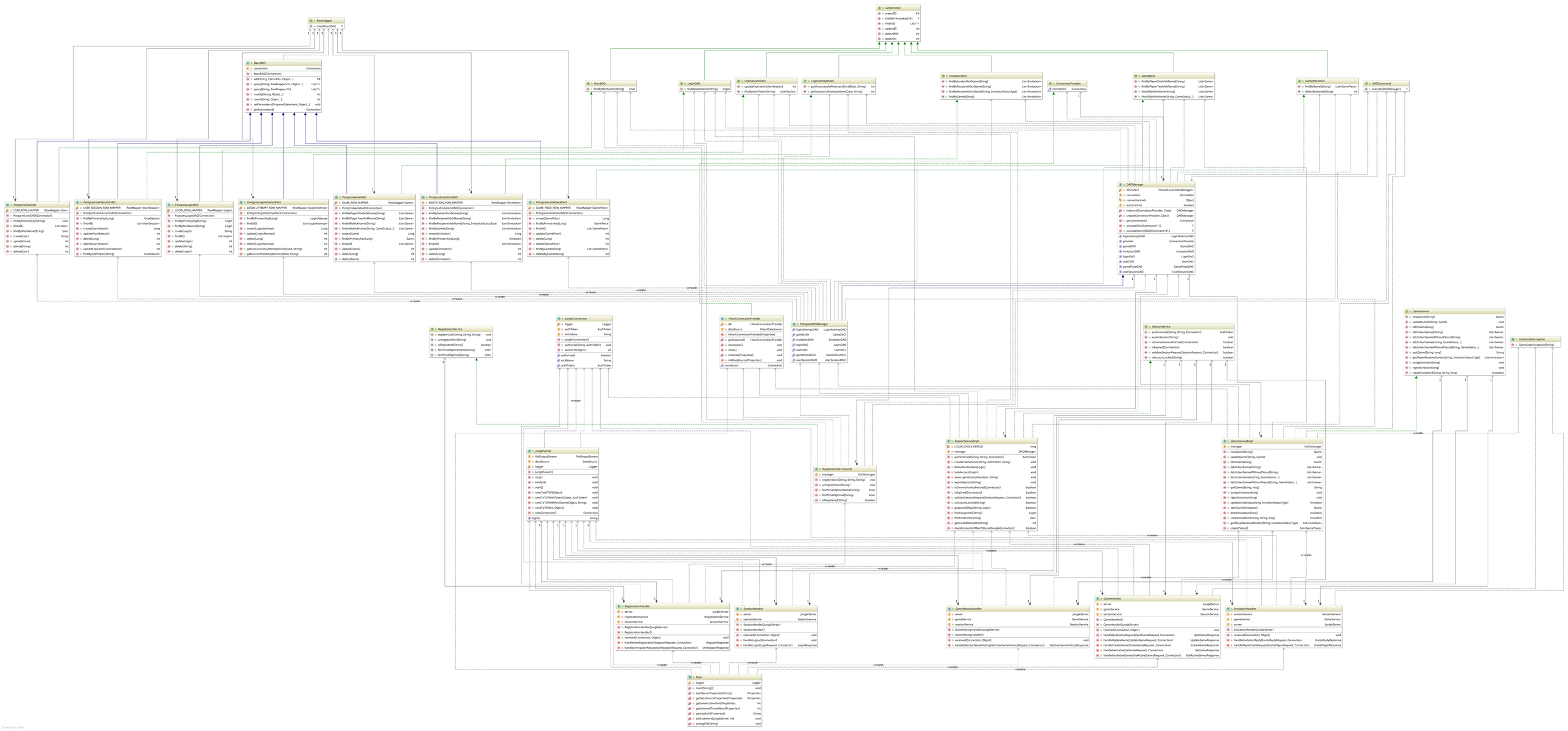
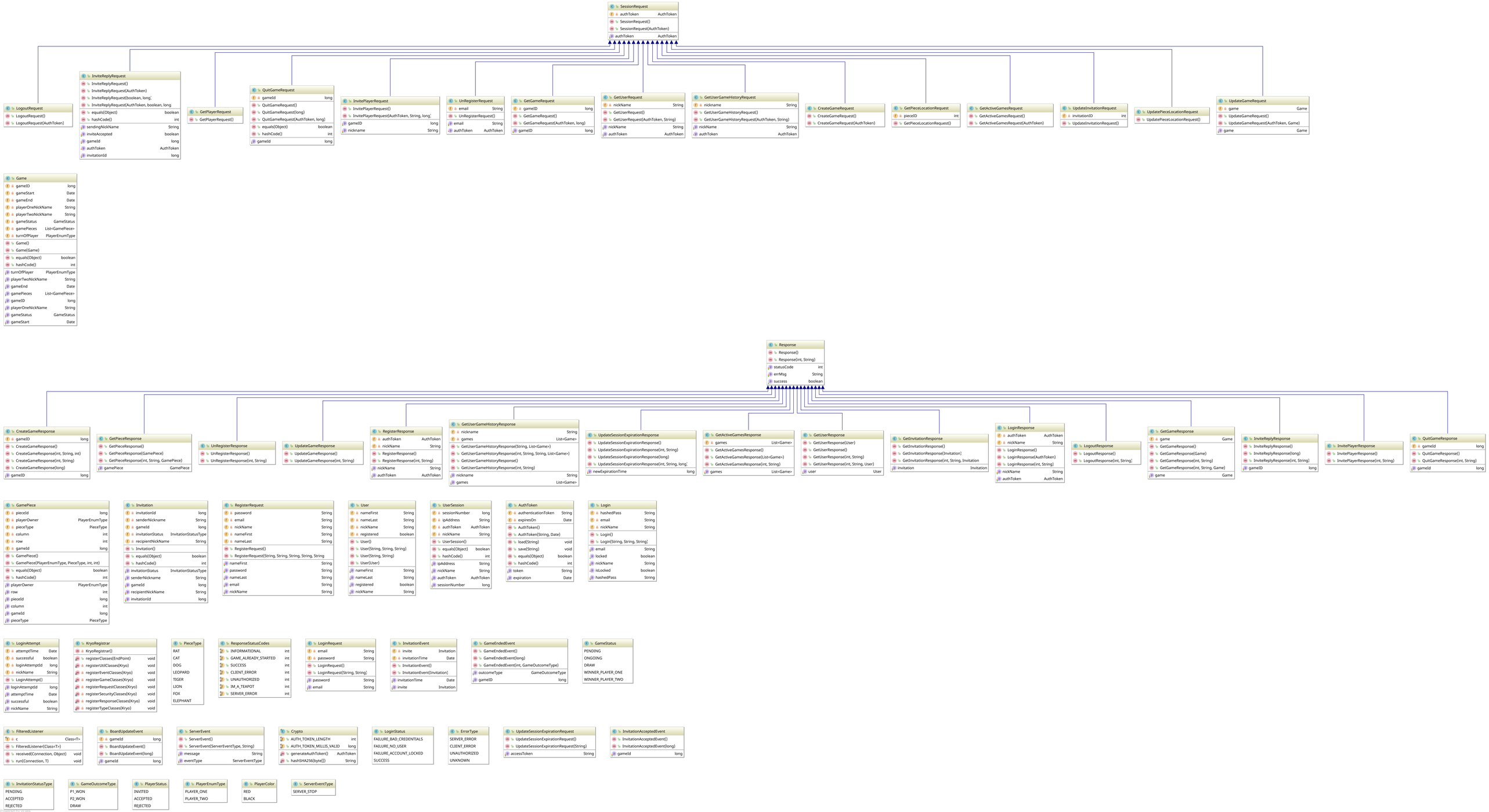

RiverSquare
DenSquare
JungleGamePiece
BoardSquare
TigerPiece
LeoparDPiece
TrapSquare
RatPiece
LionPiece
ElephantPiece
DogPiece
FoxPiece
CatPiece
JungleGamePieceFactory
GameBoard
JungleGame
Controller
View
LogListener
Move
GameHistoryModel
HomeController
LoginController
RegisterController
GameBoardController
NegaMax
MoveFinder
GamesModel
InvitesModel
AccountModel
JungleClient
InviteListCell
AlgorithmType
HomeControllerImpl
GameBoardControllerImpl
LoginControllerImpl
RegisterControllerImpl
ControllerFactory
HomeListener
BaseView
LoginListener
RegistrationListener
GameBoardViewImpl
ArtificialIntelligenceViewImpl
HomeViewImpl
RegisterViewImpl
LoginViewImpl
App
MainAI
Main
«create»


RowMapper
mapResultSet() T
BaseDAO
connection Connection
BaseDAO(Connection)
addE(String, Class<PK>, Object...) PK
query(String, RowMapper<T>, Object...) List<T>
query(String, RowMapper<T>) List<T>
modify(String, Object...) int
count(String, Object...) int
setParameters(PreparedStatement, Object...) void
getConnection() Connection
GenericDAO
create(T) PK
findByPrimaryKey(PK) T
findAll() List<T>
update(T) int
delete(PK) int
delete(T) int
UserDAO
findByNickName(String) User
LoginDAO
findByNickName(String) Login
UserSessionDAO
updateExpiration(UserSession) int
findByAuthToken(String) UserSession
LoginAttemptDAO
getUnsuccessfulAttemptsSince(Date, String) int
getSuccessfulAttemptsSince(Date, String) int
InvitationDAO
findByGenderNickName(String) List<Invitation>
findByRecipientNickName(String) List<Invitation>
findByRecipientNickName(String, InvitationStatusType) List<Invitation>
findByGameId(long) List<Invitation>
ConnectionProvider
connection Connection
GameDAO
findByPlayerOneNickName(String) List<Game>
findByPlayerTwoNickName(String) List<Game>
findByNickNameOne(String) List<Game>
findAllByNickNameOne(String, GameStatus...) List<Game>
GamePieceDAO
findByGameId(long) List<GamePiece>
deleteByGameId(long) int
DAOCommand
execute(DAOManager) T
PostgreUserDAO
USER_ROW_MAPPER RowMapper<User>
PostgreUserDAO(Connection)
findByPrimaryKey(String) User
findAll() List<User>
findByNickName(String) User
create(User) String
update(User) int
delete(String) int
delete(User) int
PostgreUserSessionDAO
USER_SESSION_ROW_MAPPER RowMapper<UserSession>
PostgreUserSessionDAO(Connection)
findByPrimaryKey(Long) UserSession
findAll() List<UserSession>
create(UserSession) Long
update(UserSession) int
delete(Long) int
delete(UserSession) int
updateExpiration(UserSession) int
findByAuthToken(String) UserSession
PostgreLoginDAO
LOGIN_ROW_MAPPER RowMapper<Login>
PostgreLoginDAO(Connection)
findByPrimaryKey(String) Login
findByNickName(String) Login
create(Login) String
findAll() List<Login>
update(Login) int
delete(String) int
delete(Login) int
PostgreLoginAttemptDAO
LOGIN_ATTEMPT_ROW_MAPPER RowMapper<LoginAttempt>
PostgreLoginAttemptDAO(Connection)
findByPrimaryKey(Long) LoginAttempt
findAll() List<LoginAttempt>
create(LoginAttempt) Long
update(LoginAttempt) int
delete(Long) int
delete(LoginAttempt) int
getUnsuccessfulAttemptsSince(Date, String) int
getSuccessfulAttemptsSince(Date, String) int
PostgreGameDAO
GAME_ROW_MAPPER RowMapper<Game>
PostgreGameDAO(Connection)
findByPlayerOneNickName(String) List<Game>
findByPlayerTwoNickName(String) List<Game>
findByNickName(String) List<Game>
findAllByNickName(String, GameStatus...) List<Game>
create(Game) Long
findByPrimaryKey(Long) Game
findAll() List<Game>
update(Game) int
delete(Long) int
delete(Game) int
PostgreInvitationDAO
INVITATION_ROW_MAPPER RowMapper<Invitation>
PostgreInvitationDAO(Connection)
findBySenderNickName(String) List<Invitation>
findByRecipientNickName(String) List<Invitation>
findByRecipientNickName(String, InvitationStatusType) List<Invitation>
findByGameId(long) List<Invitation>
create(Invitation) Long
findByPrimaryKey(Long) Invitation
findAll() List<Invitation>
update(Invitation) int
delete(Long) int
delete(Invitation) int
PostgreGamePieceDAO
GAME_PIECE_ROW_MAPPER RowMapper<GamePiece>
PostgreGamePieceDAO(Connection)
create(GamePiece) Long
findByPrimaryKey(Long) GamePiece
findAll() List<GamePiece>
update(GamePiece) int
delete(Long) int
delete(GamePiece) int
findByGameId(long) List<GamePiece>
deleteByGameId(long) int
DAOManager
INSTANCE ThreadLocal<DAOManager>
connection Connection
connectionLock Object
autoCommit boolean
instance(ConnectionProvider, Class) DAOManager
create(ConnectionProvider, Class) DAOManager
getConnection() Connection
execute(DAOCommand<T>) T
executeAttempt(DAOCommand<T>) T
loginAttemptDAO LoginAttemptDAO
provider ConnectionProvider
gameDAO GameDAO
invitationDAO InvitationDAO
loginDAO LoginDAO
userDAO UserDAO
gamePieceDAO GamePieceDAO
userSessionDAO UserSessionDAO
RegistrationService
registerUser(String, String, String) void
unregisterUser(String) void
isRegistered(String) boolean
fetchUserByNickName(String) User
fetchUserByEmail(String) User
JungleConnection
logger Logger
authToken AuthToken
nickName String
JungleConnection()
authorize(String, AuthToken) void
sendTCP(Object) int
authorized boolean
nickName String
authToken AuthToken
HikariConnectionProvider
db HikariConnectionProvider
dataSource HikariDataSource
HikariConnectionProvider(Properties)
getInstance() HikariConnectionProvider
shutdown() void
close() void
initialize(Properties) void
initDataSource(Properties) void
connection Connection
PostgreDAOManager
loginAttemptDAO LoginAttemptDAO
gameDAO GameDAO
invitationDAO InvitationDAO
loginDAO LoginDAO
userDAO UserDAO
gamePieceDAO GamePieceDAO
userSessionDAO UserSessionDAO
SessionService
authenticate(String, String, Connection) AuthToken
expireSession(String) void
isConnectionAuthorized(Connection) boolean
isAuth(Connection) boolean
validateSessionRequest(SessionRequest, Connection) boolean
isAccountLocked(String) boolean
GameService
newGame(String) Game
updateGame(String, Game) void
fetchGame(long) Game
fetchUserGames(String) List<Game>
fetchUserGamesWithoutPieces(String) List<Game>
fetchUserGames(String, GameStatus...) List<Game>
fetchUserGamesWithoutPieces(String, GameStatus...) List<Game>
quitGame(String, long) String
acceptInvitation(long) void
rejectInvitation(long) void
createInvitation(String, String, long) Invitation
getPlayerReceivedInvites(String, InvitationStatusType) List<Invitation>
acceptInvitation(long) void
rejectInvitation(long) void
createInvitation(String, String, long) Invitation
GameStateException
GameStateException(String)
JungleServer
fileOutputStream FileOutputStream
dataSource DataSource
logger Logger
JungleServer()
stop() void
bind(int) void
start() void
sendToAllTCP(Object) void
sendToTCPWithToken(Object, AuthToken) void
sendToTCPWithNickName(Object, String) void
sendToTCP(int, Object) void
newConnection() Connection
logFile String
RegistrationServiceImpl
manager DAOManager
registerUser(String, String, String) void
unregisterUser(String) void
fetchUserByNickName(String) User
fetchUserByEmail(String) User
isRegistered(String) boolean
SessionServiceImpl
LOGIN_CHECK_PERIOD long
manager DAOManager
authenticate(String, String, Connection) AuthToken
createUserSession(String, AuthToken, String) void
failAuthentication(Login) void
lockAccount(Login) void
saveLoginAttempt(Boolean, String) void
expireSession(String) void
isConnectionAuthorized(Connection) boolean
isExpired(Connection) boolean
validateSessionRequest(SessionRequest, Connection) boolean
isAccountLocked(String) boolean
passwordOkay(Login, Login) boolean
fetchLoginInfo(String) Login
fetchUserInfo(String) User
getInvalidAttempts(String) int
doesConnectionMatchStoredJungleConnection() boolean
GameServiceImpl
manager DAOManager
newGame(String) Game
updateGame(String, Game) void
fetchGame(long) Game
fetchUserGames(String) List<Game>
fetchUserGamesWithoutPieces(String) List<Game>
fetchUserGames(String, GameStatus...) List<Game>
fetchUserGamesWithoutPieces(String, GameStatus...) List<Game>
quitGame(String, long) String
acceptInvitation(long) void
rejectInvitation(long) void
updateInviteStatus(long, InvitationStatusType) void
startGame(Invitation) Game
deleteInvitation(long) Invitation
createInvitation(String, String, long) Invitation
getPlayerReceivedInvites(String, InvitationStatusType) List<Invitation>
createPieces() List<GamePiece>
RegistrationHandler
server JungleServer
registrationService RegistrationService
sessionService SessionService
RegistrationHandler(JungleServer)
RegistrationHandler()
received(Connection, Object) void
handleNewRegistration(RegisterRequest, Connection) RegisterResponse
handleUnregisterRequest(UnRegisterRequest, Connection) UnRegisterResponse
SessionHandler
server JungleServer
sessionService SessionService
SessionHandler(JungleServer)
SessionHandler()
received(Connection, Object) void
handleLogout(Connection) void
handleLogin(LoginRequest, Connection) LoginResponse
GameHistoryHandler
server JungleServer
gameService GameService
sessionService SessionService
GameHistoryHandler(JungleServer)
GameHistoryHandler()
received(Connection, Object) void
handleGetUserGameHistory(GetUserGameHistoryRequest, Connection) GetUserGameHistoryResponse
GameHandler
server JungleServer
gameService GameService
sessionService SessionService
GameHandler()
GameHandler(JungleServer)
received(Connection, Object) void
handleQuitGameRequest(QuitGameRequest, Connection) QuitGameResponse
handleUpdateGame(UpdateGameRequest, Connection) UpdateGameResponse
handleCreateGame(CreateGameRequest, Connection) CreateGameResponse
handleGetGame(GetGameRequest, Connection) GetGameResponse
handleGetActiveGames(GetActiveGameRequest, Connection) GetActiveGamesResponse
InvitationHandler
sessionService SessionService
gameService GameService
server JungleServer
InvitationHandler(JungleServer)
received(Connection, Object) void
handleInvitationReply(InviteReplyRequest, Connection) InviteReplyResponse
handlePlayerInviteRequest(InvitePlayerRequest, Connection) InvitePlayerResponse
Main
logger Logger
main(String[]) void
loadServerProperties(String) Properties
getDataSourceProperties(Properties) Properties
getServerListenPort(Properties) int
getListenerThreadNum(Properties) int
getLogPath(Properties) String
addListeners(JungleServer, int) void
setLogFile(String) void
«create»
Created by yFiles



PlantUML editor
Powered by yr-nos