

# Protocol

King of Jawa

Jannik Jaberg

22. Mai 2018



# KING OF JAWA

## **Inhaltsverzeichnis**

<b>1</b>	<b>Protocol Definitions</b>	<b>3</b>
<b>2</b>	<b>Protocol Types</b>	<b>4</b>
<b>3</b>	<b>Type Executions (Server)</b>	<b>5</b>
<b>4</b>	<b>Type Executions (Client)</b>	<b>8</b>

## 1 Protocol Definitions

Our protocol is used as a part of a two dimensional package. The package contains header information (our protocol) and data contents. The header is defined as following:

header	meaning
type	This field describes the package type. A type can be registered inside the Client/Server initialiser and must be bound to a specified manager, which is an singleton instance.
name	The name describes the specific execution, which should be done. it will be registered inside the manager which is bound to the type and bound to an PackageExecutable - implementing class.
id	package id

**Example for a chatPackage which changes the user's name:**

```
'head'|{'name'|'setName','type'|'USER','id'|'33'},'body'|{'oldName'|'hidin','newName'|'LANLord'}
```

## 2 Protocol Types

- Server
  - CONNECTION
  - CHAT
  - PING
  - HIGHSCORE
  - LOBBY
  - GAME
- Client
  - CONNECTION
  - CHAT
  - PING
  - USER
  - HIGHSCORE
  - LOBBY
  - GAME
  - SHAQ

### 3 Type Executions (Server)

- CONNECTION

name	call-reason	arguments
requestHandshake	a client wants to connect to the server	name : userName
onClientQuit	a client closed the game	none

- CHAT

name	call-reason	arguments
sendMessage	a client wants to send a message	id : the senders userId room : the chat-room the message has to be send to message : the message content

- PING

name	call-reason	arguments
ping	a client wants to know his ping	recTime : sets the received time from the sent time.
pong	the answer of a server ping	sentTime : sets the sent time.

- USER

name	call-reason	arguments
none	none	none

- LOBBY

name	call-reason	arguments
requestLobbies	a new client connects to the game and requests all existing lobbies	none
openLobbies	a client wants to open a new lobby	none
removeLobby	a client closed his lobby or disconnected while the game didn't run	lobbyId : the id of the lobby which has to be removed
removePlayer-FromLobby	removes a player from the lobby	lobbyId : the id of the lobby from which the player has to be removed.
joinLobby	execute the joinrequest for a lobby.	lobbyId : the id of the lobby from which the player has to be joined.
startLobby	executes a start-request for a lobby.	lobbyId : the id of the lobby, which has to be started. lobbyMap : the map, which has to be played.
onMapChanged	updates the currently selected map.	lobbyId : the id of the lobby, which map was updated. lobbyMap : the map, which has to be played.
onStateChange	updates the state of a specific user.	lobbyId : the id of the lobby in which a user changed his state.

- HIGHSCORE

name	call-reason	arguments
requestHighscore	a newly connected client wants to receive the highscore	none

- GAME

<b>name</b>	<b>call-reason</b>	<b>arguments</b>
requestBuild	requests a building	BuildingType : type of the building tile : the tile the building should be placed
mapLoaded	tells the server that the map on the client is loaded	lobby : the game lobby
requestBuildingInfo	requests the building-info for a specific building	entityUUID : the uuid of the building, which the info was requested for.
requestLevelUp	requests a levelup for a specific building	entityUUID : the uuid of the building the level should be increased for.

## 4 Type Executions (Client)

- CONNECTION

name	call-reason	arguments
validateHandshake	the server processed the handshake	outcome : ok   if it's okay outcome : failed   if not

- CHAT

name	call-reason	arguments
sendMessage	a client wants to send a message	id : the senders userId room : the chat-room the message has to be send to message : the message content
directMessage	there is a direct message for this specific client	id : should be none message : the direct message room : should be none
roomMessage	a message was sent to a chat-room in which the client is registered	id : should be none message : the message room : the chat-room the message has to be send to
privateMessage	a private message was sent to the client	id : should be none rId : the receipants id message : the message content room : none

- PING

name	call-reason	arguments
ping	the server wants to know the ping	none
pong	the answer of client ping	none

- USER

name	call-reason	arguments
setName	the player has a new name	oldName : the player's old name newName : the player's new name
addPlayer	a new player has to be added to the local player list	newName : the player's name isLocal : a boolean whether the player is the local client id : the players id
removePlayer	a player has to be removed off the local player list	id : the player's id



- LOBBY

name	call-reason	arguments
createLobby	creates a lobby.	lobbyId : the lobby id. lobbyName : the name of the lobby. ownerId : the ownerUUid userCount : the user-count. lobbyState : the lobby-state.
synchronizeLobby	creates a lobby.	lobbyId : the lobby id. lobbyName : the name of the lobby. ownerId : the ownerUUid userCount : the user-count. lobbyState : the lobby-state.
deleteLobby	a lobby closed and has to be removed off the local lobby list	lobbyId : the id of the lobby to be removed
onJoin	adds an user to the lobby.	lobbyId : the id of the lobby the user has to be added to. joinedUser : the user id of the joined user.
onQuit	removes an user from the lobby.	lobbyId : the id of the lobby the user has to be removed from. leftUser : the user id of the left user.
onStateChange	changes the lobbystate.	lobbyId : the id of the lobby. lobbyState : the state of the lobby.
onUserStateChange	changes a user state.	lobbyId : the id of the lobby. joinedUser : the user the state has changed for. userState : the new state.
onMapChange	changes the map preview.	lobbyId : the id of the lobby. lobbyMap : the new map.

- highscore

name	call-reason	arguments
sendHighscore	at first connection the server sends the highscore	none
refreshHighscore	the higscore has to be updated	none

- GAME

name	call-reason	arguments
initGameStart	initiates the game.	mapName : the name of the map
updateResources	updates the resources for the client.	res:"resName": the resource amount.
broadcastBuild	broadcasts a build.	buildingPosition:x : the x-position buildingPosition:y : the y-position buildingOwner : the owner id. buildingClass : the building to be built. entityUUId : the buildings uuid.
broadcast-IslandUpdate	updates an island.	islandId : the island id color : the color.
sendBuildingInfo	sends the building info for a corresponding building.	currentLevel : the current level of a building. upgradeCostCoin : the coin-cost for an upgrade. upgradeCostStone : the stone-cost for an upgrade. upgradeCostWood : the wood-cost for an upgrade. nextLevelIncomeCoin : the income per minute for the next level. nextLevelCoinCostPM : the coin-cost per minute for the next level. nextLevelIncomeWood : the income per minute for the next level. nextLevelIncomeStone : the income per minute for the next level. currentIncomeStone : the income per minute for the current level. currentIncomeWood : the income per minute for the current level. currentIncomeCoin : the income per minute for the current level. currentCoinCostPM : the current coin cost per minute. maxLevel : the maximum level.