**Statuscode-Intervalle**:   
general success: 0-50 “lobby” success: 100-150 “play” success: 200-250 “debug” success: 300-350

general error: 50-100 “lobby” error: 150-200 “play” error: 250-300 “debug” error: 350-400

**Seiten:**

“lobby” commands : S. 1-5 “play” commands: S. 6-19 “debug” commands: S. 20 “exit” command: S.19

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| \_ | \_ | “command”:   * “<undefined”   “command\_key”:   * “<undefined>” | Error:  "response\_code": 50,  "response\_msg": "Commmand <command> not found!" | Falls ein Command-Key außerhalb der definierten Command-Keys übergeben wird, Error | {"command": "xy",  "command\_key": "xy"} |
| \_ | \_ | “command”:   * “<defined”   “command\_key”:   * “<defined>” | Error:  "response\_code": 51,  "response\_msg": "Received payload is not correct JSON!" | Falls invalides JSON übergeben wird | {"command"= "xy",  "command\_key"= "xy"} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „lobby“ | „create“ | “command”:   * “lobby”   “command\_key”:   * “create” | Success: "response\_code": 100,  "response\_msg": "Lobby created!",  "key": "<lobby-key>" | Kreiert Lobby im Server | {“command”: “lobby”,  “command\_key”: “create”} |
| Error:  "response\_code": 150,  "response\_msg": "Client already in a lobby!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „lobby“ | „join“ | “command”:   * “lobby”   “command\_key”:   * “join”   “key”:   * “<lobby-key>”   “pos”:   * “p1” * “p2” * “sp” | Success: "response\_code": 101,  "response\_msg": "Joined Lobby!",  "key": "<lobby-key>" "pos": "<pos>" | Der Lobby beitreten (zweiter Spieler, Spectator) | {“command”: “lobby”,  “command\_key”: “join”,  “key”:  “<lobby-key>”} |
| Error:  "response\_code": 151,  "response\_msg": "Lobby does not exist!"  "key":  <lobby-key>  "response\_code": 150,  "response\_msg": "Client already in a lobby!"  "response\_code": 152,  "response\_msg":  "Failed to join lobby! Maybe position blocked?  "key":  <lobby-key>" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „lobby“ | „leave“ | “command”:   * “lobby”   “command\_key”:   * “leave” | Success: "response\_code": 102,  "response\_msg": "Client left lobby." | Die Lobby verlassen | {“command”: “lobby”,  “command\_key”: “leave”} |
| Error:  "response\_code": 153,  "response\_msg": "Client not in a lobby!" |
| „lobby“ | „swap“ | “command”:   * “lobby”   “command\_key”:   * “swap”   “pos”:   * “p1” * “p2” * “sp” | Success: "response\_code": 103,  "response\_msg": "Client swapped!"  "pos": "<pos>" | Die Position in der Lobby wechseln.  p1 ist jener Spieler, der den ersten Zug macht, heißt will man tauschen, wer den ersten Zug machen soll, swap benutzen | {“command”: “lobby”,  “command\_key”: “swap”, “pos”: “p2”} |
| Error:  "response\_code": 154,  "response\_msg":  "Position unknown!" "pos": "<pos>"  "response\_code": 153,  "response\_msg": "Client not in a lobby!"  "response\_code": 155,  "response\_msg":  "Position already occupied!"  "pos": "<pos>" |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „lobby“ | „pos“ | “command”:   * “lobby”   “command\_key”:   * “pos” | Success: "response\_code": 104,  "response\_msg": "Client position is:" "pos": "<pos>" | Gibt die Position eines Clients zurück (p1, p2 oder sp (spectator)) | {“command”: “lobby”,  “command\_key”: “pos”} |
| Error:  "response\_code": 153,  "response\_msg": "Client not in a lobby!" |
| „lobby“ | „status" | “command”:   * “lobby”   “command\_key”:   * “status” | Success: "response\_code": 105,  "response\_msg": "Status of lobby:",  "P1": "True",  "P2": "False",  "Spectators": 0,  "GameClient": "True",  "key": "<lobby-key" | Gibt den Status einer Lobby zurück | {“command”: “lobby”,  “command\_key”: “status”} |
| Error:  "response\_code": 153,  "response\_msg": "Client not in a lobby!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „lobby“ | \_ | “command”:   * “lobby”   “command\_key”:   * “<undefined>” | Error:  "response\_code": 50,  "response\_msg": "Commmand <command\_key> not found!" | Falls ein Command-Key außerhalb der definierten Command-Keys übergeben wird, Error | {"command": "lobby",  "command\_key": "xy"} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | \_ | “command”:   * “play”   “command\_key”:   * “<**defined**>” | Error:  "response\_code": 250,  "response\_msg": "No game client connected. Try again later!"  "response\_code": 251,  "response\_msg": "A spectator can not play!"  "response\_code": 153,  "response\_msg": "Client not in a lobby!" | Valider Command und Command-Key, aber sonstige Error, die auftreten können in Verbindung mit Command „play“ | / |
| “command”:   * “play”   “command\_key”:   * “<**undefined**>” | Error:  "response\_code": 50,  "response\_msg": "Commmand <command\_key> not found!" | Falls ein Command-Key außerhalb der definierten Command-Keys übergeben wird, Error | {"command": "play",  "command\_key": "xy"} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| / | / | / | Success: "response\_code": 202,  "response\_msg":  "Game over:",  "result": 1,  "turn": 11  *[broadcast, einmalig]*  "response\_code": 205,  "board": “[0,1,0,0]”  *[spielerspezifisch, konstant]*  "response\_code": 206,  "representation": <konsolenausgabe\_board>  *[broadcast, konstant]*  Image  <image>  *[broadcast, konstant]* | Wenn im “game loop” seiend, wird initial beim Anlegen des Spiels und nach jedem Zug [broadcast] bzw. nach jedem Zug des Gegners [spielerspezifisch] dieses Set an Nachrichten rausgeschickt. Wenn das Spiel vorbei ist, einmalig ein „Game Over“ | / |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „create“ | “command”:   * “play”   “command\_key”:   * “create”   “game”:   * “connect4” * “tictactoe” * “othello" * “nim” * “checkers” * “go” * “waldmeister”   “mode”:   * “player\_vs\_player” * “player\_vs\_ai” * “playerai\_vs\_ai” * “playerai\_vs\_playerai”   “difficulty”:   * “easy” * “medium” * “hard” | Success:  "response\_code": 200,  "response\_msg":  "Game initialized" | Initialisiert das Spiel im Backend mit der übergebenen Spielkonfiguration | {"command": "play",  "command\_key": "create",  "game": "connect4",  "mode": "player\_vs\_ai",  "difficulty":  "hard"} |
| Error:  "response\_code": 253,  "response\_msg": "Arguments are missing or invalid!",  "mode":  "p",  "difficulty":  "hard"  "response\_code": 254,  "response\_msg": "Game still running. Please surrender first!"  "response\_code": 52,  "response\_msg": "Internal error occurred!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „valid\_moves“ | “command”:   * “play”   “command\_key”:   * “valid\_moves”   *Bei Spielen mit Spielsteinen verschieben zusätzlich die from\_pos:*  “pos”:   * “<int>” | Success:  "response\_code": 208,  "response\_msg": "Valid moves:",  "moves": "[1, 2, 5]"  Image  <image> | Zeigt valide Züge.  Rückgabe als Bild mit gehighlighteten Feldern und als String mit validen Zugindizes. | {"command": "play",  "command\_key": "valid\_moves",  "pos":  "<from\_pos>"} |
| Error: "response\_code": 252,  "response\_msg": "You need to create a game first!"  *[initial und wenn gegenwärtiges Spiel beendet]*  "response\_code": 257,  "response\_msg": "Pos ‘<pos>’ is not a pos!",  "pos": "<pos>"  "response\_code": 257,  "response\_msg": "Pos must be greater than or equal to 0!"  "response\_code": 257,  "response\_msg": "Invalid from\_pos!" |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „make\_move“ | “command”:   * “play”   “command\_key”:   * “make\_move”   “move”:   * “<move>”   + *Spiele, bei denen nur gesetzt wird: to\_pos:  „<int >“*   + *Spiele, bei denen gezogen wird from\_pos und to\_pos als Tupel:  „(<int>, <int>)“* | Success:  "response\_code": 207,  "response\_msg":  "Valid move." | Führt Zug aus | {"command": "play",  "command\_key": "make\_move",  "move":  "(<from\_pos>, <to\_pos>)"} |
| Error:  "response\_code": 252,  "response\_msg": "You need to create a game first!"  *[initial und wenn gegenwärtiges Spiel beendet]*  "response\_code": 255,  "response\_msg":  "’move’ entry not set!"  "response\_code": 256,  "response\_msg":  "Invalid move!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „undo\_move“ | “command”:   * “play”   “command\_key”:   * “undo\_move”   “num”:   * “<int>” | Success:  "response\_code": 209,  "response\_msg":  "Move successfully undone." | <num> Züge rückgängig machen. | {"command": "play",  "command\_key": "undo\_move”,  "num":  "<num >"} |
| Error:  "response\_code": 252,  "response\_msg": "You need to create a game first!"  "response\_code": 258,  "response\_msg": "Amount of moves to be undone not declared!"  "response\_code": 259,  "response\_msg": "Num: ‘<num>’ is not an int!"  "response\_code": 259,  "response\_msg": " Amount of moves to be undone must be greater than 0!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „surrender“ | “command”:   * “play”   “command\_key”:   * “surrender” | Success: "response\_code": 210,  "response\_msg":  "Game over:" "result": "-1" | Stoppt laufendes Spiel und löscht dessen Spielinformation im Backend, darauffolgend dann entweder command\_key „quit“ oder „new\_game“ | {"command": "play",  "command\_key": "surrender"} |
| Error:  "response\_code": 252,  "response\_msg": "You need to create a game first!"  *[initial und wenn gegenwärtiges Spiel beendet]* |
| „play“ | „quit“ | “command”:   * “play”   “command\_key”:   * “quit” | Success: "response\_code": 211,  "response\_msg": "Game quit." | Fährt Docker-Container herunter, trennt Verbindung von Server zu GameClient | {"command": "play",  "command\_key": "quit"} |
| Error:  "response\_code": 254,  "response\_msg": "Game still running. Please surrender first!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „new\_game“ | “command”:   * “play”   “command\_key”:   * “new\_game” | Success: "response\_code": 200,  "response\_msg":  "Game initialized" | Initialisiert neues Spiel im Backend (in noch aktivem Docker-Container, mit aktiver GameClient-Verbindung) | {"command": "play",  "command\_key": "new\_game "} |
| Error:  "response\_code": 252,  "response\_msg": "You need to create a game first!"  "response\_code": 254,  "response\_msg": "Game still running. Please surrender first!"  "response\_code": 52,  "response\_msg": "Internal error occurred!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „blunder“ | “command”:   * “play”   “command\_key”:   * “blunder” | Success: "response\_code": 212,  "response\_msg": "No obvious blunder.”  "response\_code": 213,  "response\_msg": "Blunder list (index, move):",  "blunder": "[(0, 0), (8, 0)]" | Zeigt Züge, die nicht gut waren.  Funktion für nach Spielende | {"command": "play",  "command\_key": "blunder"} |
| Error:  "response\_code": 252,  "response\_msg": "You need to create a game first!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „timeline“ | “command”:   * “play”   “command\_key”:   * “timeline”   “num”:   * “<int>” | Success:  "response\_code": 214,  "representation": <konsolenausgabe\_board>  Image <image> | Spiel nach Spielende noch einmal anschauen unter Angabe einer Zugnummer <num>.  Wert 0 entspricht dem ersten Zug. | {"command": "play",  "command\_key": "timeline ",  "num":  "<num>"} |
| Error: "response\_code": 252,  "response\_msg": "You need to create a game first!"  "response\_code": 260,  "response\_msg": "Timeline start index not declared!"  "response\_code": 261,  "response\_msg": "Index: ‘<num>’ is not an int!"  "response\_code": 261,  "response\_msg": "Index must be greater than or equal to 0!"  "response\_code": 261,  "response\_msg": "Invalid timeline index!" |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „step“ | “command”:   * “play”   “command\_key”:   * “step” | Success:  "response\_code": 214,  "representation": <konsolenausgabe\_board>  Image <image> | Innerhalb der Timeline um einen Zug nach vorne springen | {"command": "play",  "command\_key": "step"} |
| Error: "response\_code": 252,  "response\_msg": "You need to create a game first!"  "response\_code": 261,  "response\_msg": "Invalid timeline index!" |
| „play“ | „unstep“ | “command”:   * “play”   “command\_key”:   * “unstep” | Success:  "response\_code": 214,  "representation": <konsolenausgabe\_board>  Image <image> | Innerhalb der Timeline um einen Zug nach hinten springen | {"command": "play",  "command\_key": "unstep"} |
| Error: "response\_code": 252,  "response\_msg": "You need to create a game first!"  "response\_code": 261,  "response\_msg": "Invalid timeline index!" |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „evaluate“ | “command”:   * “play”   “command\_key”:   * “evaluate”   “game”:   * “connect4” * “tictactoe” * “othello" * “nim” * “checkers” * “go” * “waldmeister”   “difficulty”:   * “easy” * “medium” * “hard”   “num”   * “<int >” | Success:  "response\_code": 201,  "response\_msg": "Evaluation runs"  "response\_code": 203,  "response\_msg": "Evaluation finished:",  "wins": 3,  "losses": 1,  "draws": 0 | <num> Spiele (Maximum: 100) spielen lassen ohne Anzeige. Statistische Auswertung über Anzahl Siege, Anzahl Niederlagen, und Anzahl Unentschieden.  Möglichkeit, beim ersten der n Spiele sich Blunder und Timeline anzeigen zu lassen | {"command": "play",  "command\_key": "evaluate",  "game": "connect4",  "difficulty":  "hard", “num”: “<num>”} |
| Error:  "response\_code": 254,  "response\_msg": "Game still running. Please surrender first!"  "response\_code": 262,  "response\_msg": "Num of games at evaluation not declared!"  "response\_code": 263,  "response\_msg": "Num: ‘<num>’ is not an int!"  "response\_code": 263,  "response\_msg": "1 or more than 100 games not supported at evaluation!"  "response\_code": 253,  "response\_msg": "Arguments are missing or invalid!",  "difficulty":  "hard"  "response\_code": 52,  "response\_msg": "Internal error occurred!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „play“ | „stop\_evaluate“ | “command”:   * “play”   “command\_key”:   * “stop\_evaluate” | Success:  "response\_code": 203,  "response\_msg": "Evaluation finished:",  "wins": 3,  "losses": 1,  "draws": 0 | Unterbrechung der Evaluation mit statistischer Auswertung der bisherig vollständig gespielten Spiele | {"command": "play",  "command\_key": "stop\_evaluate"} |
| Error:  "response\_code": 252,  "response\_msg": "You need to create a game first!" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „exit“ | / | “command”:   * “exit” | / | Trennt die Verbindung vom Client zum Server | {"command": "exit"} |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| command | command\_key | Parameter | Rückgabewert | Beschreibung | Beispiel-Request |
| „debug“ | „active\_container“ | “command”:   * “debug”   “command\_key”:   * “active\_container“ | "0": {  …  }  "1": {  …  } "response\_code": 300,  "response\_msg": "List of active GameClients:",  "count": 2, | Entspricht docker container ls  (Liste aller aktiven Container) | {"command": "debug",  "command\_key": "active\_container "} |
| „debug“ | „toggle\_game\_client  \_debug“ | “command”:   * “debug”   “command\_key”:   * “toggle\_game\_client   \_debug“ | "response\_code": 301,  "response\_msg": "GameClient debug toggled.",  "debug": true | Hält Docker-GameClient-Container offen, auch bei Fehlereintritt, wenn der GameClient intern geschlossen wird | {"command": "debug",  "command\_key": "toggle\_game  \_client\_debug"} |