|  |  |  |  |
| --- | --- | --- | --- |
| **Pfad** | **Request-Körper** | **Response** | **Beschreibung** |
| http://localhost:4567/games/hangman/ |  |  |  |
| start/neuerNutzer | Name des Nutzers mit key „name“ | True, wenn anlegen des neuen Nutzers erfolgreich, sonst false | Zu Beginn gibt sich der Nutzer einen Namen über den er im Spiel referenziert werden kann. Dazu muss dieser Name dem Server übergeben werden und dort gespeichert werden. |
| start/neuesWort/0 |  |  |  |
| start/neuesWort/1 |  |  |  |
| start/neuerPool | 'name' = Nutzername, 'pool' = Pool-ID und 'level' = Schwierigkeitsgrad werden übergeben | True, wenn erstellen des neuen Pools erfolgreich war, false sonst | Wenn ein neuer Pool angeglegt wird, muss der Nutzer diesem eine ID geben. Wenn diese bereits vergeben ist, kann der Pool nicht angelegt werden. Sonst ist der Nutzer automatisch Mitglied im Pool und wartet auf einen Gegner |
| start/poolSuchen | 'pools angefragt‘ | True, wenn kein Pool vorhanden ist, false sonst | Wenn Nutzer einem Pool beitreten möchte, wird zunächst überprüft, welche Pools vorhanden sind. Ist kein Pool vorhanden, muss zunächst einer erstellt werden. Wenn ein oder mehrere Polls vorhanden sind, werden diese ausgegeben |
| start/beitreten | 'name' = Nutzername, 'pool' = wunschID werden übergeben | True, wenn Nutzer erfolgreich dem Pool beigetreten, false sonst | Wenn der Nutzer einem bestimmten Pool beitreten möchte, muss er die gewünschte Pool-ID übergeben. Wenn der Beitritt erfolgreich ist, kommt der Nutzer sofort in den Warteraum des entsprechenden Pools |
|  |  |  |  |
|  |  |  |  |

//Rateversuche  
 *post*("/games/hangman/start/neuesWort/0", (q, a) -> {return RespClass.*buchstabeRaten*(q.body());}); //Buchstabe raten  
 *post*("/games/hangman/start/neuesWort/1", (q, a) -> {return RespClass.*wortRaten*(q.body());}); //Wort raten  
  
  
 //Menueoptionen  
 *post*("/games/hangman/start/neuerPool/", (q, a) -> { return RespClass.*neuerPool*(q.body());}); //neuen Pool anlegen  
 *post*("/games/hangman/start/poolSuchen/", (q, a) -> { return RespClass.*getPools*();}); // alle Pools schicken  
 *post*("/games/hangman/start/meinePools/", (q, a) -> { return RespClass.*meinePools*(q.body());}); //individuelle Pools schicken  
 *post*("/games/hangman/start/beitreten/", (q, a) -> {return RespClass.*poolBeitreten*(q.body());}); //Pools beitreten  
  
 //WarteRaum im Pool  
 *post*("/games/hangman/start/pool/warteRaum", (q,a) -> {return RespClass.*poolWarteRaum*(q.body());});  
  
 //Spielstatus abfragen/antwort --> wie dieses polling aufm blatt  
 *post*("/games/hangman/start/spiel/status", (q,a) -> {return RespClass.*status*(q.body());});  
 //Abfrage , wer als erstes anfangen darf  
 *post*("/games/hangman/start/spiel/anfang", (q,a) -> {return RespClass.*anfang*(q.body());});  
  
// post("/games/hangman/start/lobby",(q, a)-> LogikIntern.lobby(q.body()));