

Allgemeines

- Server URL <https://rbsg.uniks.de>.
- Das Registrieren und Einloggen am Server sind die einzigen Aktionen, die ohne Login funktionieren.
- Basis URL für REST <https://rbsg.uniks.de/api>.
- Basis URL für WS <wss://rbsg.uniks.de/ws>.
- Alle Anfragen müssen als JSON gesendet werden, Ausnahme ist die *noop* (no operation) Anweisung für Asynchrone Verbindungen.
- Der Server antwortet immer im JSON Format.
- REST Anfragen die ein Login voraussetzen, benötigen einen zusätzlichen Header *userKey*. Der Server erstellt bei jedem Login ein neuen Key und invalidiert den alten.
- Nutzer werden nach 15 min. automatisch ausgeloggt, sofern keine Aktion ausgeführt wird.
- Spiele werden nach 5 min. gelöscht, sofern diese nicht korrekt gestartet werden.
- Asynchrone Verbindungen müssen regelmäßig benutzt werden, da ansonsten die Verbindung wegen eines Timeouts geschlossen wird.
- Alle Asynchronen Verbindungen akzeptieren den *noop* Befehl. Dieser kann als Klartext ohne Format gesendet werden und dient lediglich zum Erhalt der Verbindung.

Synchron REST (<https://rbsg.uniks.de/api>)

POST /user

Beschreibung:

Registriert einen neuen Nutzer am Server. Name und Passwort müssen einzigartig sein.

Body:

- name: String
- password: String

GET /user

Beschreibung:

Liefert eine Liste aller Benutzer, die sich in der Lobby befinden.

POST /user/login

Beschreibung:

Führt ein Login durch.

Body:

- name: String
- password: String

GET /user/logout

Beschreibung:

Führt ein Logout aus.

POST /game

Beschreibung:

Erzeugt ein neues Spiel. Mit einem optionalen Seed kann eine Karte wiederverwendet werden.

Body:

- name: String
- neededPlayer: Integer
- seed: Long (optional)

GET /game

Beschreibung:

Liefert eine Liste aller Spiele zurück.

GET /game/:id(?spectator=true)

Beschreibung:

Tritt dem Spiel mit der gegebenen ID bei.

Query Parameter:

- spectator: boolean (optional)

Parameter:

- id: String

DELETE /game/:id

Beschreibung:

Löscht das Spiel mit der gegebenen ID.

Parameter:

- id: String

POST /army

Beschreibung:

Erzeugt eine neue eigene Armee.

Body:

- name: String
- units: String[] (Liste von Einheiten-IDs)

GET /army

Beschreibung:

Liefert eine Liste aller eigener Armeen.

GET /army/:id

Beschreibung:

Liefert eine eigene Armee mit der gegebenen ID.

Parameter:

- id: String

PUT /army/:id

Beschreibung:

Updated eine eigene Armee mit der gegebenen ID.

Parameter:

- id: String

Body:

- name: String
- units: String[] (Liste von Einheiten-IDs)

DELETE /army/:id

Beschreibung:

Löscht eine eigene Armee mit der gegebenen ID.

Parameter:

- id: String

GET /army/units

Beschreibung:

Liefert eine Liste aller verfügbaren Einheitentypen.

Asynchron WS (<wss://rbsg.uniks.de/ws>)

/system

Beschreibung:

Auf diesem Channel werden System Nachrichten versendet. Zum Beispiel, ein Spieler hat die Lobby betreten / verlassen oder ein Spiel wurde erzeugt / gelöscht.

Auf diesen Channel können keine Nachrichten gesendet werden.

/chat?user=MyUserName

Beschreibung:

Auf diesem Channel werden Chat Nachrichten gesendet und verteilt. Es gibt Öffentliche und Private Chats.

Query Parameter:

- user: String

Befehle:

- Nachricht an alle
 - channel: String (all)
 - message: String
- Nachricht an einen bestimmten Spieler
 - channel: String (private)
 - to: String
 - message: String

/game?gameId=123456789(&armyId=987654321)(& spectator=true)

Beschreibung:

Auf diesem Channel werden Spiel relevante Nachrichten gesendet. Zur Anmeldung wird eine Spiel-ID benötigt. Optional kann eine Armee zum antreten mitgegeben werden. Des Weiterem ist es möglich als Beobachter beizutreten, in diesem Modus können keine Aktionen ausgeführt werden.

Query Parameter:

- gameId: String
- armyId: String (Optional)
- spectator: boolean (Optional)

Befehle:

- Kommandos (Spiellobby)
 - messageType: String (command)
 - Armee auswählen
 - action: String (changeArmy)
 - data: String (ID einer eigenen Armee)
 - Spiel verlassen
 - action: String (leaveGame)
 - Spieler bereit
 - action: String (readyToPlay)
 - Spiel starten
 - action: String (startGame)
- Kommandos (Laufendes Spiel)
 - messageType: String (command)
 - Spiel verlassen
 - action: String (leaveGame)

- Einheit bewegen
 - action: String (moveUnit)
 - data:
 - unitId: String (Id der Einheit, die bewegt werden soll)
 - path: String[] (Liste von Feld-Ids, die einen Pfad bildet)

- Einheit angreifen
 - action: String (attackUnit)
 - data:
 - unitId: String (Id der Einheit, die angreifen soll)
 - toAttackId: String (Id der Einheit, die angegriffen werden soll)

- In die nächste Phase eintreten
 - action: String (nextPhase)

- Chat
 - messageType: String (chat)
 - Nachricht an alle
 - channel: String (all)
 - message: String

 - Nachricht an einen bestimmten Spieler
 - channel: String (private)
 - to: String
 - message: String

Der Ablauf für ein Spiel sieht wie folgt aus:

- Spieler treten einem Spiel bei.
- Spieler wählt eine Armee (Erst bei Spielbeginn für alle sichtbar).
 - Das Wählen einer Armee setzt einen Spieler auf „nicht bereit“.
- Spieler setzt sich selbst als „Bereit“.
- Ein beliebiger Spieler startet das Spiel. (Sofern alle bereit sind)
- Spielerarmeen werden platziert.
- Spieler sind nacheinander an der Reihe.
- Ein Spielzug besteht aus 3 Phasen.
 - Bewegen. (Mindestens eine Einheit muss bewegt werden)
 - Angreifen. (Kann übersprungen werden)
 - Restlichen Bewegungspunkte verwenden. (Kann übersprungen werden)
- Alle Einheiten können nur benachbarte feindliche Einheiten angreifen.
- Der letzte Spieler mit einer lebenden Einheit hat gewonnen.
- Alle Partien sind Deathmatches.