















# Battalion 1944 RCon API Documentation

Bulkhead Interactive

October 19, 2018

## Contents

<b>1</b>	<b>Overview</b>	<b>3</b>
<b>2</b>	<b>Sending A Request</b>	<b>3</b>
<b>3</b>	<b>Global Errors</b>	<b>3</b>
<b>4</b>	<b>Connection Request</b>	<b>4</b>
4.1	Example Request Payload . . . . .	4
4.2	Required Fields . . . . .	4
4.3	Optional Fields . . . . .	4
4.4	Example Response Payload(s) . . . . .	4
<b>5</b>	<b>Ping Server</b>	<b>5</b>
5.1	Example Request Payload . . . . .	5
5.2	Required Fields . . . . .	5
5.3	Optional Fields . . . . .	5
5.4	Example Response Payload(s) . . . . .	5
<b>6</b>	<b>Get Mod List</b>	<b>6</b>
6.1	Example Request Payload . . . . .	6
6.2	Required Fields . . . . .	6
6.3	Example Response Payload(s) . . . . .	6
<b>7</b>	<b>Player Sync </b>	<b>7</b>
7.1	Example Request Payload . . . . .	7
7.2	Required Fields . . . . .	7
7.3	Example Response Payload(s) . . . . .	7
<b>8</b>	<b>Game Sync </b>	<b>8</b>
8.1	Example Request Payload . . . . .	8
8.2	Required Fields . . . . .	8
8.3	Example Response Payload(s) . . . . .	8
<b>9</b>	<b>Config Sync </b>	<b>9</b>
9.1	Example Request Payload . . . . .	9
9.2	Required Fields . . . . .	9
9.3	Optional Fields . . . . .	9
9.4	Example Response Payload(s) . . . . .	9
<b>10</b>	<b>Blacklist Player </b>	<b>10</b>
10.1	Example Request Payload . . . . .	10
10.2	Required Fields . . . . .	10
10.3	Example Response Payload(s) . . . . .	10

<b>11 UnBlacklist Player</b> 	<b>11</b>
11.1 Example Request Payload . . . . .	11
11.2 Required Fields . . . . .	11
11.3 Example Response Payload(s) . . . . .	11
<b>12 Announce</b> 	<b>12</b>
12.1 Example Request Payload . . . . .	12
12.2 Required Fields . . . . .	12
12.3 Example Response Payload(s) . . . . .	12
<b>13 Auto Balance</b> 	<b>13</b>
13.1 Example Request Payload . . . . .	13
13.2 Required Fields . . . . .	13
13.3 Example Response Payload(s) . . . . .	13
<b>14 Disconnect All</b> 	<b>14</b>
14.1 Example Request Payload . . . . .	14
14.2 Required Fields . . . . .	14
14.3 Example Response Payload(s) . . . . .	14
<b>15 Kick Player</b> 	<b>15</b>
15.1 Example Request Payload . . . . .	15
15.2 Required Fields . . . . .	15
15.3 Example Response Payload(s) . . . . .	15
<b>16 Load Next Game</b> 	<b>16</b>
16.1 Example Request Payload . . . . .	16
16.2 Required Fields . . . . .	16
16.3 Example Response Payload(s) . . . . .	16
<b>17 Restart</b> 	<b>17</b>
17.1 Example Request Payload . . . . .	17
17.2 Required Fields . . . . .	17
17.3 Example Response Payload(s) . . . . .	17
<b>18 State Change</b> 	<b>18</b>
18.1 Example Request Payload . . . . .	18
18.2 Required Fields . . . . .	18
18.3 Example Response Payload(s) . . . . .	18
<b>19 List Log Files</b> 	<b>19</b>
19.1 Example Request Payload . . . . .	19
19.2 Required Fields . . . . .	19
19.3 Example Response Payload(s) . . . . .	19
<b>20 Get Log File</b> 	<b>20</b>
20.1 Example Request Payload . . . . .	20
20.2 Required Fields . . . . .	20
20.3 Example Response Payload(s) . . . . .	20
<b>21 Delete Log File</b> 	<b>21</b>
21.1 Example Request Payload . . . . .	21
21.2 Required Fields . . . . .	21
21.3 Example Response Payload(s) . . . . .	21

# 1 Overview

In an attempt to make server administration or Battalion 1944 easier, we have developed a remote connection (RCon) API that allows developers to build their own tools. We are also in the progress of developing our own tool that will be available soon.

This API is currently under development, and we welcome requests for functionality. Please email [api@bulkheadinteractive.com](mailto:api@bulkheadinteractive.com) with any feature requests.

## 2 Sending A Request

**Structure:** `http://<IP>:<PORT>`


**Verb:** POST

**Content-Type Header:** `application/json`

**Body:** Request JSON payload, described in each following sections.

All requests are sent as a JSON payload to the IP and Port of the RCon connection. The RCon port is always the game port + 2. For example, a game port of 7777 will have an RCon port of 7779. This is forced within the game code and is not changeable by server admins. Please remember to forward this port to allow RCon requests to be received and fielded by the game server.

All requests require a “RequestType” field that matches the titled section. For example, a connection request requires a “RequestType” field of “ConnectionRequest”.

Some requests will require a password field, which must match the “RConPassword” defined in the server’s DefaultGame.ini. It is strongly recommended that this password is set. Failure to do so by a server administrator will open these (usually secured) requests up and allow anyone with the correct IP and Port to call them. All secured requests will be denoted by a  symbol and will have the “Password” field as a requirement.

**Note:** There is an **8KiB** (8192 byte) limit on the receive of the request payload. Going over this limit will simply not return anything, as the server will only receive the last 8KiB of the payload. This is particularly relevant for *Config Sync*, as you cannot send the entire config back. You must only send the changes you want to make.

## 3 Global Errors

There are three global errors that can be returned due to either an incorrect password, missing password or the game having an invalid GameMode. The 2 former are due to user error on the request. The latter should never happen, but is there for safety and indicates some kind of failed server start.

---

```
{
  "error": "This request is password protected, but you have provided an incorrect password."
}
```

---

```
{
  "error": "This request is password protected, but you have provided no password."
}
```

---

```
{
  "error": "This request requires use of the GameMode, but no valid GameMode is currently
    accessible."
}
```

---

## 4 Connection Request

This is called when a client uses a 'Connect' command inside their game client. There is no reason to call this manually at any other time.

### 4.1 Example Request Payload

---

```
{  
  "RequestType": "ConnectionRequest",  
  "ConnectionPassword": "Password",  
  "TeamToJoin": "1"  
}
```

---

### 4.2 Required Fields

- **RequestType** - This must match "ConnectionRequest".

### 4.3 Optional Fields

- **ConnectionPassword** - this is the password to join the game, if there is one. This is returned in the response payload so that it can be automatically appended to the connection command inside the client.
- **TeamToJoin** - this is the team you wish to join. If none is specified, you will be asked to join a team when connecting to the server. This is returned in the response payload so that it can be automatically appended to the connection command inside the client.

### 4.4 Example Response Payload(s)

---

```
{  
  "SteamID": "90114349769462794",  
  "Port": 8000,  
  "Map": "Upottery",  
  "ConnectionPassword": "Password",  
  "TeamToJoin": "1"  
}
```

---

## 5 Ping Server

This is called when a client is on the server browser page. It is used to get a ping time to the server in question. This can also be used as a 'handshake' to check a server can be communicated with.

### 5.1 Example Request Payload

---

```
{  
  "RequestType": "PingServer",  
  "IP": "127.0.0.1",  
  "Port": 7777,  
  "SentTimestamp": 1530023661194  
}
```

---

### 5.2 Required Fields

- **RequestType** - This must match "PingServer".

### 5.3 Optional Fields

- **IP** - The IP of this server.
- **Port** - The port of this server.
- **SentTimestamp** - The ticks are the time of the sending.

### 5.4 Example Response Payload(s)

---

```
{  
  "SentTimestamp": 1530023661194  
}
```

---

## 6 Get Mod List

This request can be used to see what mods a server has installed, along with information about each mod.

### 6.1 Example Request Payload

---

```
{
  "RequestType": "GetModList",
  "Password": "test"
}
```

---

### 6.2 Required Fields

- **RequestType** - This must match “GetModList”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini

### 6.3 Example Response Payload(s)

---

```
{
  "currentMapModId": "",
  "allModsIds": [
    "1393428529"
  ],
  "bIncludesStandalonePackages": true,
  "installedModPackages": [
    {
      "modPath": "/MountainTown/",
      "steamWorkshopPublishedId": "1393428529",
      "bContentOnly": true,
      "modTypes": [
        "Map"
      ],
      "weapons": [],
      "maps": [
        "Bergstadt_Background1",
        "Bergstadt_2_Spawns",
        "Bergstadt_2_BombMode",
        "Bergstadt_2",
        "Bergstadt_2_Blockout"
      ],
      "author": "",
      "version": 1,
      "versionName": "1.0",
      "description": ""
    }
  ]
}
```

---

## 7 Player Sync

This can be called to request a list of all players in a server, and which teams they belong to.

### 7.1 Example Request Payload

---

```
{
  "RequestType": "PlayerSync",
  "Password": "Password"
}
```

---

### 7.2 Required Fields

- **RequestType** - This must match “PlayerSync”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini

### 7.3 Example Response Payload(s)

---

```
{
  "PlayerData": {
    "teams": [
      {
        "players": [
          "76561198369254052",
          "76561198368551426"
        ]
      },
      {
        "players": [
          "76561198369738648",
          "76561198368607090",
          "76561198321458528"
        ]
      }
    ],
    "registeredPlayers": [
      "76561198368551426",
      "76561198321458528",
      "76561198369254052",
      "76561198369738648",
      "76561198368607090",
      "76561198369030716"
    ]
  }
}
```

---

## 8 Game Sync

This request can be called to gather information about an in-progress game. The response for this will always contain *Wartide* specific information, which can be ignored for other modes.

### 8.1 Example Request Payload

---

```
{
  "RequestType": "GameSync",
  "Password": "Password"
}
```

---

### 8.2 Required Fields

- **RequestType** - This must match “GameSync”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini

### 8.3 Example Response Payload(s)

---

```
{
  "PlayerData": {
    "teams": [
      {
        "score": 0,
        "nation": "Nation_US",
        "name": "Allies",
        "winStreak": 0,
        "loseStreak": 0,
        "kills": 0,
        "deaths": 0,
        "tokens": 5,
        "tokensEarnedThisRound": 0,
        "numTeamMembers": 0
      },
      {
        "score": 0,
        "nation": "Nation_DE",
        "name": "Axis",
        "winStreak": 0,
        "loseStreak": 0,
        "kills": 0,
        "deaths": 0,
        "tokens": 5,
        "tokensEarnedThisRound": 0,
        "numTeamMembers": 0
      }
    ],
    "map": "Liberation",
    "mode_Short": "WRT",
    "mode_Long": "Wartide",
    "steamLobbyId": "90114556452825094"
  }
}
```

---



## 9 Config Sync

This request can be used to both gather and modify the state of the servers' variables. It first gathers the default DefaultGame.ini's variables, then reads any modification sent in the request along with any already saved on the server. It saves the changes sent to the file stored on the server and sends the whole thing back. That way, changes sent in the request are saved and applied over the top of the DefaultGame.ini, allowing the server to be reset at any moment by deleting the *rcon/config/rcon-config.config* file.

### 9.1 Example Request Payload

---

```
{
  "RequestType": "ConfigSync",
  "Password": "Password",
  "config_items": {
    "items": [
      {
        "name": "ServerName",
        "value": "SERVER",
        "type": "string"
      }
    ]
  }
}
```

---

### 9.2 Required Fields

- **RequestType** - This must match “ConfigSync”.
- **Password** - This must match the “RConPassword” in the servers' loaded DefaultGame.ini

### 9.3 Optional Fields

- **Config\_items** - An array of config items to be changed.

### 9.4 Example Response Payload(s)

---

```
{
  "config_items": {
    "items": [
      {
        "name": "NumInitialTokensTeam",
        "value": "0",
        "type": "int"
      },
      {
        "name": "NumInitialTokensPlayer",
        "value": "1",
        "type": "int"
      }
    ]
  }
}
```

---

## 10 Blacklist Player

This request can be used to add a players' SteamID64 to the list of blacklisted players. The blacklist is a list of players who are banned from the server while their IDs are present within. The list is loaded on server launch, and added to using this request.

**Note:** This will not kick a player from a server, it will only prevent them joining it again. Please use ***KickPlayer*** to remove them from the server.

**Note:** This request cannot remove an ID from the blacklist.

### 10.1 Example Request Payload

---

```
{
  "RequestType": "BlacklistPlayer",
  "Password": "Password",
  "PlayerIDs": [
    "700000000000000001",
    "700000000000000002",
    "700000000000000003"
  ]
}
```

---

### 10.2 Required Fields

- **RequestType** - This must match “BlacklistPlayer”.
- **Password** - This must match the “RConPassword” in the servers' loaded DefaultGame.ini
- **PlayerIDs** - JSON array of SteamID64 in string form of the players' you want to add to the blacklist

### 10.3 Example Response Payload(s)

---

```
{
  "info": "Successfully updated blacklist with players' IDs."
}
```

---

## 11 UnBlacklist Player

This request can be used to remove a players' SteamID64 to the list of blacklisted players. This is the only way to remove an item from the blacklist remotely. You cannot add and remove in one request.

**Note:** This request cannot add an ID to the blacklist.

### 11.1 Example Request Payload

---

```
{
  "RequestType": "UnBlacklistPlayer",
  "Password": "Password",
  "PlayerIDs": [
    "700000000000000001",
    "700000000000000002",
    "700000000000000003"
  ]
}
```

---

### 11.2 Required Fields

- **RequestType** - This must match “UnBlacklistPlayer”.
- **Password** - This must match the “RConPassword” in the servers' loaded DefaultGame.ini
- **PlayerIDs** - JSON array of SteamID64 in string form of the players' you want to remove from the blacklist

### 11.3 Example Response Payload(s)

---

```
{
  "info": "3 players' IDs successfully removed from the blacklist."
}
```

---

---

```
{
  "error": "Warning! No IDs were removed from the Blacklist. They may not have been present in the first place."
}
```

---

## 12 Announce

This request can be used to remotely invoke an announcement on a server instance. An announcement will appear both in chat and on the players' screens. This is useful for announcement of maintenance, updates, restarts or config changes. It could also be used to announce the next map and mode from a rotation.

**Note:** As the announcement appears on players' screens, it is recommended you keep announcements to a minimum during play.

### 12.1 Example Request Payload

---

```
{
  "RequestType": "Announce",
  "Password": "Password",
  "Announcement": "This server will restart in 5 minutes for routine maintenance"
}
```

---

### 12.2 Required Fields

- **RequestType** - This must match "Announce".
- **Password** - This must match the "RConPassword" in the servers' loaded DefaultGame.ini
- **Announcement** - A string field with the announcement string to be shown to players.

### 12.3 Example Response Payload(s)

---

```
{
  "info": "Successfully made announcement."
}
```

---

---

```
{
  "error": "Announcement field was empty."
}
```

---

## 13 Auto Balance

This request can be used to remotely invoke a Auto Balance command on a server instance. This can be used on a timer to keep long games balanced.

### 13.1 Example Request Payload

---

```
{  
  "RequestType": "Autobalance",  
  "Password": "Password",  
}
```

---

### 13.2 Required Fields

- **RequestType** - This must match “Autobalance”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini

### 13.3 Example Response Payload(s)

---

```
{  
  "info": "Auto balancing now."  
}
```

---

## 14 Disconnect All

This request can be used to remotely kick all players in a server.

**Note:** There is a delay of around 0.5 seconds from the kick being received and it happening.

### 14.1 Example Request Payload

---

```
{
  "RequestType": "DisconnectAll",
  "Password": "Password"
}
```

---

### 14.2 Required Fields

- **RequestType** - This must match “DisconnectAll”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini

### 14.3 Example Response Payload(s)

---

```
{
  "info": "Kicking all now."
}
```

---

---

```
{
  "error": "Error disconnecting all players. There may have been no-one in the server to disconnect."
}
```

---

## 15 Kick Player

This request can be used to remotely kick a select player or list of players from a server. It will not ban them, so it is recommended you first ***Blacklist*** a player if you wish to do that.

### 15.1 Example Request Payload

---

```
{
  "RequestType": "KickPlayer",
  "Password": "Password",
  "PlayerIDs": [
    "700000000000000001",
    "700000000000000002"
  ]
}
```

---

### 15.2 Required Fields

- **RequestType** - This must match “KickPlayer”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini
- **PlayerIDs** - JSON array of SteamID64 in string form of the players’ you want to kick from the server

### 15.3 Example Response Payload(s)

---

```
{
  "info": "Kicking 2 player(s) now."
}
```

---

---

```
{
  "error": "Error kicking players. There may have been no-one in the server with the supplied
           player IDs."
}
```

---

## 16 Load Next Game

This request can be used to remotely load the next game in the rotation. Rotations are cyclical, so this will restart the rotation if you reach the end of a defined rotation. If there is only one Map and Mode combination defined in the rotation, it will act the same as a ***Restart*** command, loading the same game up again.

### 16.1 Example Request Payload

---

```
{
  "RequestType": "LoadNextGame",
  "Password": "Password"
}
```

---

### 16.2 Required Fields

- **RequestType** - This must match “LoadNextGame”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini

### 16.3 Example Response Payload(s)

---

```
{
  "info": "Loading next game now."
}
```

---

---

```
{
  "error": "Can't load next game yet as there are still players travelling."
}
```

---



## 17 Restart

This request can be used to remotely restart the current game.

### 17.1 Example Request Payload

---

```
{
  "RequestType": "Restart",
  "Password": "Password"
}
```

---

### 17.2 Required Fields

- **RequestType** - This must match “Restart”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini

### 17.3 Example Response Payload(s)

---

```
{
  "info": "Restarting game now."
}
```

---

## 18 State Change

This request can be used to change the in-game state of a server. The states are as follows:

- **WaitingToStart** - This state waits to meet the start conditions before progressing.
- **CountdownToStart** - After meeting the start conditions, countdown to begin the game.
- **SetupRound** - The buy phase during all game modes before the round start.
- **RoundInProgress** - The game in progress. CTF, DOM and TDM are considered single round modes, so this will conclude the game for these modes.
- **PostRound** - Display which team won the round. Safest way to end a round early.
- **PostMatch** - Display who won the game after all rounds have concluded. This will gracefully end the game, showing everyone the scoreboard. Safest way to end a game early.
- **GameEnding** - End the game immediately. This is a harsh and abrupt ending.

### 18.1 Example Request Payload

---

```
{  
    "RequestType": "ChangeState",  
    "Password": "Password",  
    "NewState": "PostRound"  
}
```

---

### 18.2 Required Fields

- **RequestType** - This must match “ChangeState”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini
- **NewState** - The state you want to change to

### 18.3 Example Response Payload(s)

---

```
{  
    "info": "State successfully changed to PostRound."  
}
```

---

```
{  
    "error": "FakeState is not a valid state to change to."  
}
```

---

## 19 List Log Files

This request can be used to list all the Game Mode log files that currently exist on a server. These will only be generated and accessible if both the **LoggingEnabled** config variable is set to true and the **logLocation** config variable is pointing to a valid log location.

### 19.1 Example Request Payload

---

```
{
  "RequestType": "ListLogFiles",
  "Password": "Password"
}
```

---

### 19.2 Required Fields

- **RequestType** - This must match “ChangeState”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini

### 19.3 Example Response Payload(s)

---

```
{
  "Files": [
    "Battalion_2018.05.09-14.26.00_Liberation_Wartide.txt",
    .
    .
    .
    "Battalion_2018.05.09-14.52.40_Upottery_BASE MODE.json"
  ]
}
```

---

---

```
{
  "error": "No files located in the log file directory. Is logging enabled?"
}
```

---

---

```
{
  "error": "Could not find a log file path defined."
}
```

---

## 20 Get Log File

This request can be used to get the contents of a specific log file. You need to supply the name of the file, so it is best practice to send a **ListLogFiles** request first to get all the possible file names. This will return an array of strings, each element being a line from the log file.

### 20.1 Example Request Payload

---

```
{
  "RequestType": "GetLogFile",
  "Password": "Password",
  "LogFileName": "Battalion_2018.05.09-14.26.00_Liberation_Wartide.txt"
}
```

---

### 20.2 Required Fields

- **RequestType** - This must match “ChangeState”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini
- **LogFileName** - This must be a valid log file name

### 20.3 Example Response Payload(s)

---

```
{
  "FileContents": [
    "Match State changed to: Entering Map",
    "Match State changed to: Leaving Map"
  ]
}
```

---

```
{
  "error": "Could not find a log file path defined."
}
```

---

## 21 Delete Log File

This request can be used to delete a specific log file. You need to supply the name of the file, so it is best practice to send a *ListLogFiles* request first to get all the possible file names.

### 21.1 Example Request Payload

---

```
{
  "RequestType": "DeleteLogFile",
  "Password": "Password",
  "LogFileName": "Battalion_2018.05.09-14.26.00_Liberation_Wartide.txt"
}
```

---

### 21.2 Required Fields

- **RequestType** - This must match “ChangeState”.
- **Password** - This must match the “RConPassword” in the servers’ loaded DefaultGame.ini
- **LogFileName** - This must be a valid log file name

### 21.3 Example Response Payload(s)

---

```
{
  "info": "Successfully deleted file Battalion_2018.05.09-14.26.00_Liberation_Wartide.txt"
}
```

---

---

```
{
  "error": "Failed to delete file Battalion_2018.05.09-14.26.00_Liberation_Wartide.txt"
}
```

---

---

```
{
  "error": "File Battalion_2018.05.09-14.26.00_Liberation_Wartide.txt does not exist."
}
```

---

---

```
{
  "error": "Could not find a log file path defined."
}
```

---