

# Verwaltungsserver

---

Über den Verwaltungsserver können Spieleserver gefunden werden, die momentan online sind (indem sie stets mit dem Verwaltungsserver in Verbindung bleiben) und neue Spieler akzeptieren (es findet keine Runde auf dem Spieleserver statt).

Der Verwaltungsserver hört stets auf folgendem Port: **51963**.

Folgende Kommandos und Antworten sind für den Verwaltungsserver definiert:

## Einen Spieleserver registrieren

---

**Bezeichner des Kommandos:** `msRegisterGameServer`

**Verwendung durch:** Spieleserver

**Antwort:** keine

Dieses Kommando registriert einen Spieleserver an einem Verwaltungsserver. Idealerweise registriert sich der Spieleserver selbst mit seinen Verbindungsdaten am Verwaltungsserver. Ein Spieleserver ist über einen *TCP-Port* sowie über eine *IPv4-Adresse* oder eine *IPv6-Adresse* oder einen *Hostnamen* erreichbar, wobei diese Informationen den Spielern versteckt werden dürfen. Den Spielern wird aber stets der Name, den der Serveradministrator dem Spieleserver zugewiesen hat, angezeigt.

Der TCP-Port kann beliebig mit 2 Ausnahmen gesetzt werden. Die Ausnahmen sind 0 (in TCP/IP als reserviert gekennzeichnet) und 51963 (da für den Verwaltungsserver verwendet). Es ist Abstand von Ports zu nehmen, die für bekannte Internetdienste verwendet werden, z.B. 21 (FTP), 80 (HTTP) oder 443 (HTTPS).

Sieht die Adresse des Spieleservers wie eine IP-Adresse (v4 oder v6) aus, ist aber keine gültige IP-Adresse, so wird das Kommando vom Server verworfen.

Pro Verbindung zum Verwaltungsserver kann nur ein einziger Spieleserver registriert werden. Ein existierender Spieleserver für eine Verbindung wird überschrieben.

## Zusätzlich übermittelte Daten ( `content` )

- `serverName` (String): Ein beliebiger Name, der den Spielern zur Identifizierung des Servers angezeigt wird. Er darf nicht leer sein.
- `serverAddress` (String): Adresse (IPv4, IPv6 oder Hostname), unter der der Spieleserver erreichbar ist.
- `serverPort` (Integer): Port zur Adresse, unter der Spieleserver erreichbar ist. Wertebereich: 1-65535, ohne 51963.

## Beispiele

Beispiel 1: Wir möchten einen Spieleserver namens "Feuerland" registrieren, der unter einer IPv4-Adresse auf dem Port 20000

erreichbar ist:

```
{
  "command": "msRegisterGameServer",
  "content": {
    "serverName": "Feuerland",
    "serverAddress": "192.168.0.10",
    "serverPort": 20000
  }
}
```

Beispiel 2: Wir möchten einen Spieleserver namens "Server von Bösewicht" registrieren, der unter einer IPv6-Adresse (lokales Präfix) auf dem Port 30000 erreichbar ist:

```
{
  "command": "msRegisterGameServer",
  "content": {
    "serverName": "Server von Bösewicht",
    "serverAddress": "fd40:9dc7:b528::1",
    "serverPort": 30000
  }
}
```

Beispiel 3: Wir möchten einen Spieleserver namens "轰炸机人" registrieren, der unter einem Hostnamen auf dem Port 40000 erreichbar ist:

```
{
  "command": "msRegisterGameServer",
  "content": {
    "serverName": "轰炸机人",
    "serverAddress": "bombergame.example.org",
    "serverPort": 40000
  }
}
```

## Zustand eines Spieleservers aktualisieren

**Bezeichner des Kommandos:** `msUpdateGameServerStats`

**Verwendung durch:** Spieleserver

**Antwort:** keine

Dieses Kommando aktualisiert Informationen, die dem Verwaltungsserver über den momentanen Zustand des Spieleservers bekannt sind. Diese Informationen lauten:

- Wie viele Spieler sind auf dem Server angemeldet?
- Wie viele Spieler sind in einer Runde zulässig?
- Ist die Lobby des Servers offen oder findet gerade eine Spielerunde statt?
- Welcher Spielmodus ist aktiv?

Die Informationen werden bei einer erneuten Anforderung durch die Clients an diese übertragen.

Der Spieleserver muss vorher am Verwaltungsserver angemeldet sein. Ansonsten wird dieses Kommando durch den Verwaltungsserver verworfen.

### Zusätzlich übermittelte Daten ( `content` )

- `players` (Objekt):
  - `current` (Integer): Anzahl der momentan auf dem Server angemeldeten Spieler. Diese darf größer als die maximal zulässige Spielerzahl sein, um anzudeuten, dass weitere Spieler in der Warteschlange stehen.

- `max` (Integer): Anzahl der maximal in einer Runde zugelassenen Spieler. Diese beträgt 2, 3 oder 4.
- `isLobbyOpen` (Boolean): Ist dieser Wert `true`, so ist die Lobby des Spieleservers offen. Damit können sich neue Spieler anmelden. Ansonsten ist dieser Wert `false`.
- `gameplayMode` (Integer): Dieser Wert ist 1, wenn der auf dem Spieleserver aktive Spielmodus "Standard A" ist, oder 2, wenn der aktive Spielmodus "Standard A+B" ist.

## Beispiel

Beispiel: Auf dem Spieleserver sind 2 Spieler angemeldet. Für die nächste Runde sind 4 Spieler zugelassen, die Lobby ist momentan offen, und der Spielmodus ist "Standard A+B":

```
{
  "command": "msUpdateGameServerStats",
  "content": {
    "players": { "current": 2, "max": 4 },
    "isLobbyOpen": true,
    "gameplayMode": 2
  }
}
```

## Einen Spieleserver deregistrieren

**Bezeichner des Kommandos:** `msUnregisterGameServer`

**Verwendung durch:** Spieleserver

**Antwort:** keine

Dieses Kommando deregistriert einen eventuell registrierten Spieleserver an einem Verwaltungsserver. Da nur ein Spieleserver pro Verbindung zulässig ist und der Verwaltungsserver jeder Verbindung einen Spieleserver zuweisen kann, muss der zu deregistrierende Spieleserver nicht übermittelt werden. (Damit ist es nicht möglich, fremde Spieleserver zu deregistrieren.)

### Zusätzlich übermittelte Daten ( `content` )

Keine.

## Beispiel

Beispiel: Wir möchten unseren Spieleserver deregistrieren:

```
{
  "command": "msUnregisterGameServer"
}
```

## Liste registrierter Spieleserver anfordern

**Bezeichner des Kommandos:** `msQueryGameServers`

**Verwendung durch:** Client

**Antwort:** [Liste registrierter Spieleserver](#)

Dieses Kommando weist den Verwaltungsserver an, eine Liste mit aktuell an ihm registrierten Spieleservern an den Sender zu übermitteln.

### Zusätzlich übermittelte Daten ( `content` )

Keine.

## Beispiel

Beispiel: Wir möchten eine Liste von Spieleservern anfordern, die momentan am Verwaltungsserver registriert sind:

```
{
  "command": "msQueryGameServers"
}
```

Der Server sendet eine [Antwort](#), dessen Format in den Beispielen abgebildet ist.

## Liste registrierter Spieleserver

**Bezeichner der Antwort:** `msRQueryGameServers`

**Verwendung durch:** Client

**Antwort auf:** [Liste registrierter Spieleserver anfordern](#)

Der Verwaltungsserver sendet eine Liste von aktuell an ihm registrierten Spieleservern an den Sender, wenn er diese angefordert hat. Die Reihenfolge, in der die Server gelistet sind, ist nicht definiert.

### Zusätzlich übermittelte Daten ( `content` )

- `servers` (Array): Liste von mindestens 0 Objekten, die am Verwaltungsserver registrierte Spieleserver enthält. Jedes Objekt hat folgende Form:
  - `name` (String): Name, der den Spielern zur Identifizierung des Spieleservers angezeigt wird. Er muss identisch mit dem Namen sein, der dem Server durch das Kommando [Einen Spieleserver registrieren](#) zugewiesen wurde.
  - `address` (String): IPv4-Adresse, IPv6-Adresse oder Hostname, unter dem der Spieleserver erreicht werden kann.
  - `port` (Integer): Port, unter dem der Spieleserver an seiner Adresse erreicht werden kann. Wertebereich: 1-65535, ohne 51963.
  - `players` (Objekt):
    - `current` (Integer): Anzahl der momentan auf dem Server angemeldeten Spieler. Diese darf größer als die maximal zulässige Spielerzahl sein, um anzudeuten, dass weitere Spieler in der Warteschlange stehen.
    - `max` (Integer): Anzahl der maximal in einer Runde zugelassenen Spieler. Diese beträgt 2, 3 oder 4.
  - `isLobbyOpen` (Boolean): Ist dieser Wert `true`, so ist die Lobby des Spieleservers offen. Damit können sich neue Spieler anmelden. Ansonsten ist dieser Wert `false`.
  - `gameplayMode` (Integer): Dieser Wert ist 1, wenn der auf dem Spieleserver aktive Spielmodus "Standard A" ist, oder 2, wenn der aktive Spielmodus "Standard A+B" ist.

## Beispiele

Beispiel 1: Am Verwaltungsserver sind keine Spieleserver registriert:

```
{
  "command": "msRQueryGameServers",
  "content": {
    "servers": []
  }
}
```

Beispiel 2: Am Verwaltungsserver ist ein einziger Spieleserver registriert. An diesem ist ein Spieler angemeldet:

```

{
  "command": "msRQueryGameServers",
  "content": {
    "servers": [
      { "name": "Feuerland", "address": "192.168.0.10", "port": 20000,
        "players": { "current": "1", "max": 4 }, "isLobbyOpen": true,
        "gameplayMode": 1 }
    ]
  }
}

```

Beispiel 2: Am Verwaltungsserver sind drei Spieleserver registriert:

```

{
  "command": "msRQueryGameServers",
  "content": {
    "servers": [
      { "name": "轰炸机人", "address": "bombergame.example.org", "port": 40000,
        "players": { "current": "22", "max": 4 }, "isLobbyOpen": false,
        "gameplayMode": 2 },
      { "name": "Server von Bösewicht", "address": "fd40:9dc7:b528::1", "port": 30000,
        "players": { "current": "3", "max": 3 }, "isLobbyOpen": false,
        "gameplayMode": 1 },
      { "name": "Feuerland", "address": "192.168.0.10", "port": 20000,
        "players": { "current": "1", "max": 4 }, "isLobbyOpen": true,
        "gameplayMode": 1 }
    ]
  }
}

```