

## TeamSpeak Permission:

German

English

## Inhalt:

IDEE

TEXTBASIERTES BEISPIEL

QUERY BEFEHLE

## Idee:

---

Meiner Meinung bietet TeamSpeak ein sehr ordentliches und vielseitiges Rechtssystem, in dem neuen Client gefällt mir das „Card System“ besonders gut. Jedoch finde ich es als Entwickler schade, dass man noch keine eigenen Rechte hinzufügen kann, was mir persönlich jedoch einiges leichter machen würde.

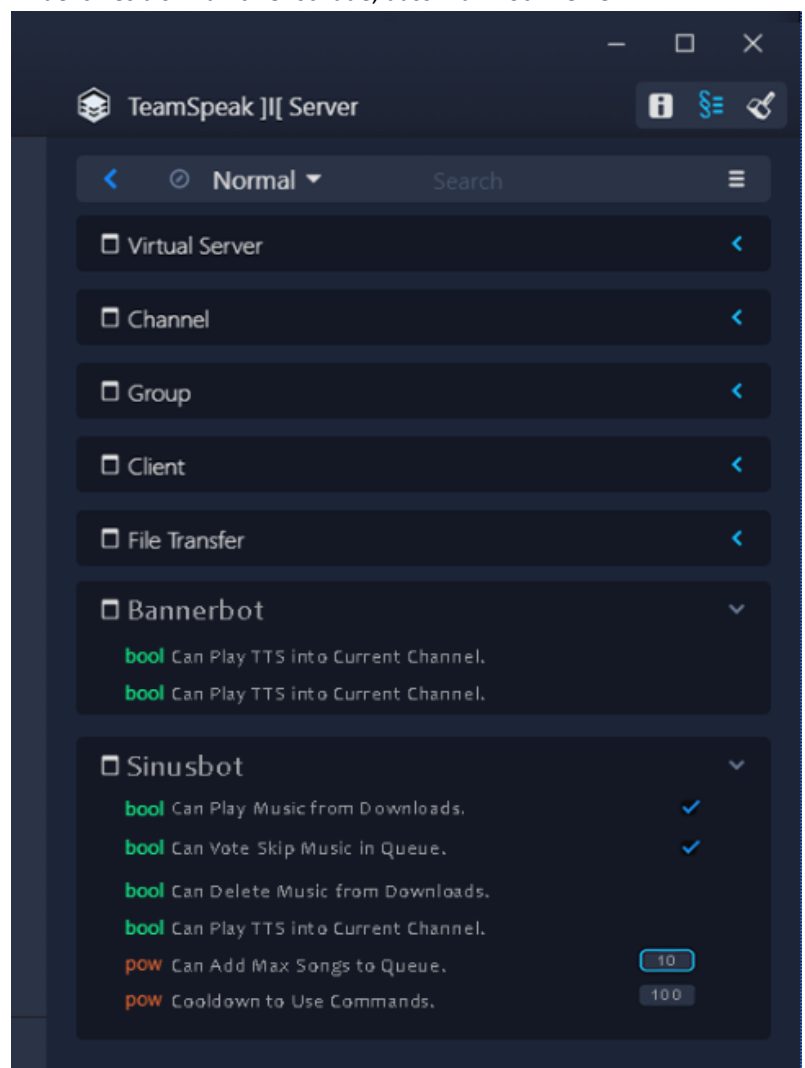
So muss man meist ein eigenes Rechtssystem in seinen Applikationen schreiben, was selbst für den Endverbraucher sehr nervig sein kann. Nehmen wir als Beispiel einen Musik Bot, wie den bekannten „Sinusbot“, bei welchen wir mit einem zweiten Rechtssystem arbeiten und etwa einen Benutzer oder eine Servergruppe als Admin festlegen müssen, damit wir den Bot richtig nutzen können.

Mein Wunsch bzw. Vorschlag wäre es eine API einzufügen, welche über die generelle „Serverquery“ Schnittstelle ansteuerbar ist. Hierbei sollte es möglich sein, seine Applikation in der Serverseitigen Datenbank zu registrieren.

Meine Idee ist es den Bot bzw. die Applikation mit einem Schlüssel in der Datenbank des Servers zu speichern. Der Schlüssel dient rein für die Zuordnung und stellt keine Rechte Funktion da.

Die Applikation kann nun entweder an eine Gruppe oder einen Client gebunden und angepasst werden.

So kann zum Beispiel die Gruppe **Normal** nur Music abspielen sowie die Music überspringen. Die Gruppe **Mod** kann dafür Music abspielen, überspringen und löschen.



So könnte man den „Sinusbot“ auf seinem Server registrieren, und somit sicherstellen, dass die Bots sich mit der Rechte Kategorie „Sinusbot“ identifizieren können.

Damit könnte man mehrere Instanzen mit nur einer Kategorie steuern.

Das Ganze sollte dynamisch aufgebaut werden, dass man zu jeder Kategorie seine Rechte Arrays mit Daten hinzufügen kann.

Ich weiß, dass dafür einige Dinge am TeamSpeak Grundsystem geändert werden müssten, aber ich denke dieses Feature wäre diese Arbeit wert. Denn letztendlich bietet diese Funktion eine lokale und vor allem Einheitliche und Übersichtliche Einbindung für Entwickler und Endverbraucher.

## Textbasiertes Beispiel:

---

### Normal:

- Channel
- Group
- Client
- File Transfer
- Sinusbot
  - o Kann Music abspielen
  - o Kann Music mit einer Abstimmung überspringen

### Mod:

- Channel
- Group
- Client
- File Transfer
- Sinusbot
  - o Kann Music abspielen
  - o Kann Music löschen
  - o Kann Music überspringen
  - o ...

### Admin:

- Channel
- Group
- Client
- File Transfer
- Bannerbot
  - o Kann Banner updaten
  - o Kann Banner löschen
- Sinusbot
  - o Kann Music abspielen
  - o Kann Music löschen
  - o Kann Music überspringen
  - o Kann Music herunterladen
  - o Kann TTS senden
  - o ...

## Query Befehle:

---

Die Query Befehle könnte man auch an das Rechte System anpassen, ich habe mir diese Befehle nur zur Demonstration ausgedacht.

- applicationadd
  - Fügt eine Applikation hinzu und gibt als Response den Applikation Schlüssel zurück.
- applicationremove
  - Löscht eine Applikation über den Schlüssel oder die ID sowie dessen Rechte und Kategorie.
- applications
  - Listet Alle verfügbaren Applikationen auf und gibt diese inklusive Schlüssel zurück.
- applicationaddperm
  - Fügt ein Element in das Kategorien Array mit den Einstellungen (bool,num,pow...)

# Content:

IDEA

TEXT BASED EXAMPLE

QUERY COMMANDS

## Idea:

---

In my opinion, TeamSpeak offers a very neat and versatile permissions system, and I particularly like the "Card system" in the new client. However, as a developer, I think it's a shame that you can't add your own permissions yet, which would make things a lot easier for me personally.

So you usually have to write your own permissions system in your applications, which can be very annoying even for the end user. Let's take a music bot as an example, like the well-known "Sinusbot", in which we work with a second permissions system and have to define a user or a server group as admin so that we can use the bot properly.

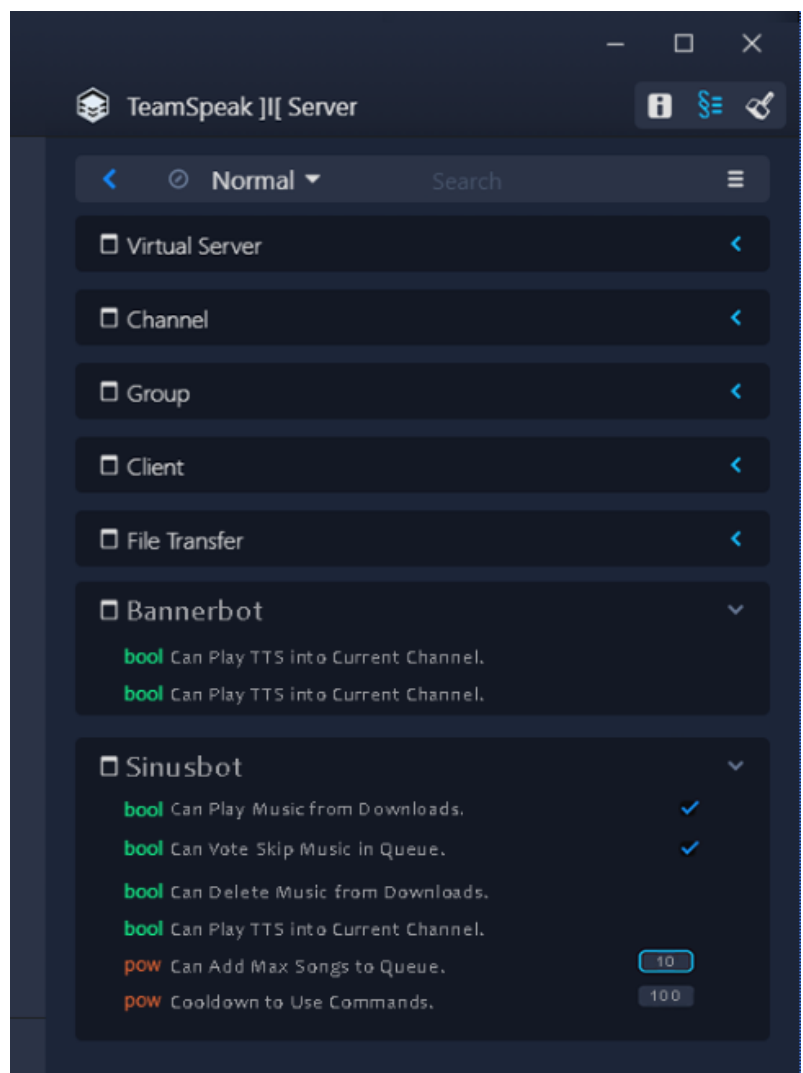
My wish or suggestion would be to add an API that can be controlled via the general "server query" interface. It should be possible to register your application in the server-side database.

My idea is to save the bot or the application with a key in the server's database. The key is used purely for the assignment and does not represent a permissions function.

The application can now either be linked to a group or a client and adapted.

For example, the Normal group can only play music and skip the music. The Mod group can play, skip and delete music for this.

So, you could register the "Sinusbot" on your server, and thus ensure that the bots can identify themselves with the permissions category "Sinusbot".



This could be used to control multiple instances with just one category. The whole thing should be built dynamically so that you can add your permissions arrays with data to each category.

I know that some things would have to be changed in the TeamSpeak basic system for this, but I think this feature would be worth the work. Because ultimately this function offers a local and, above all, uniform and clear integration for developers and end users.

## Text Based Example:

---

### Normal:

- Channel
- Group
- Client
- File Transfer
- Sinusbot
  - o Can play Music
  - o Can vote skip Music

### Mod:

- Channel
- Group
- Client
- File Transfer
- Sinusbot
  - o Can play Music
  - o Can delete Music
  - o Can skip Music
  - o ...

### Admin:

- Channel
- Group
- Client
- File Transfer
- Bannerbot
  - o Can update Banner
  - o Can delete Banner
- Sinusbot
  - o Can play Music
  - o Can delete Music
  - o Can skip Music
  - o Can download Music
  - o Can send TTS
  - o ...

## Query Commands:

---

The query commands could also be adapted to the permission system, I came up with these commands just for demonstration.

- applicationadd
  - o Adds a Applikation to the Server and return the Application Key.
- applicationremove
  - o Deletes an application using the key or ID as well as its permissions and category.
- applications
  - o Lists all available applications and returns these including keys.
- applicationaddperm
  - o Adds an element to the category array with the settings (bool, num, pow...)