

## Allgemeines

- Server URL <https://ac.uniks.de>.
- Das Registrieren und Einloggen am Server sind die einzigen Aktionen, die ohne Login funktionieren.
- Basis URL für REST <https://ac.uniks.de/api>.
- Basis URL für WS <wss://ac.uniks.de/ws>.
- Basis URL für UDP: cranberry.uniks.de:33100
- Alle Anfragen müssen als JSON gesendet werden, Ausnahme ist die *noop* (no operation) Anweisung für Asynchrone Verbindungen.
- Der Server antwortet immer im JSON Format, Ausnahme sind binäre Daten wie Bilder.
- 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.
- 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://ac.uniks.de/api>)

## POST /users

### Beschreibung:

Registriert einen neuen Nutzer am Server. Der Name muss einzigartig sein.

### Body:

- name: String
- password: String

### Beispiel:

```
> curl -X POST https://ac.uniks.de/api/users -d '{ "name": "Mr", "password": "Spock" }'
```

### Antwort:

```
{  
  "status": "success",  
  "message": "User created",  
  "data": {}  
}
```

## GET /users

### Beschreibung:

Liefert eine Liste aller Benutzer, die allgemein online sind.

### Beispiel:

```
> curl -X GET -H "userKey: xxxxx" https://ac.uniks.de/api/users
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": [  
    {  
      "id": "5e2ffbdabg75dd077d03df505",  
      "name": "kali",  
      "description": "Spielt gerade AmongUs"  
    }  
  ]  
}
```

## POST /users/login

### Beschreibung:

Führt ein Login durch.

### Body:

- name: String
- password: String

### Beispiel:

```
> curl -X POST https://ac.uniks.de/api/users/login -d '{ "name": "Mr", "password": "Spock" }'
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "userKey": "c653b568-d987-4331-8d62-26ae617847bf"  
  }  
}
```

## POST /users/logout

### Beschreibung:

Führt ein Logout aus.

### Beispiel:

```
> curl -X POST -H "userKey: xxxxx" https://ac.uniks.de/api/users/logout
```

### Antwort:

```
{  
  "status": "success",  
  "message": "Logged out",  
  "data": {}  
}
```

## POST /users/temp

### Beschreibung:

Erstellt einen temporalen Benutzer. Dieser läuft nach 24 Stunden aus.

### Beispiel:

```
> curl -X POST https://ac.uniks.de/api/users/temp
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "name": "Hendry Bracken",
    "password": "Hendry Bracken"
  }
}
```

## POST /users/:id/description

### Beschreibung:

Ändert den Beschreibungstext des eigenen Benutzers. Der Beschreibungstext ist ein Freitextfeld in das maximal 120 Zeichen geschrieben werden darf. Der Text bleibt bis zur nächsten Änderung bestehen.

### Parameter:

- id: String

### Body:

- text: String

### Beispiel:

```
> curl -X POST https://ac.uniks.de/api/users/5e2ffbdabg75dd077d03df505/description
-d '{"text": "Spielt gerade AmongUs" }'
```

### Antwort:

```
{
  "status": "success",
  "message": "Description changed",
  "data": {}
}
```

## POST /servers

### Beschreibung:

Erzeugt einen neuen Server mit einer Kategorie die einen Textkanal enthält. Der Kanal ist für alle einsehbar.

### Body:

- name: String

### Beispiel:

```
> curl -X POST -H "userKey: xxxxx" https://ac.uniks.de/api/servers  
-d '{"name": "TestServer" }'
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "id": "5e2ffbd8770dd077d03df505",  
    "name": "TestServer"  
  }  
}
```

## POST /servers/:id/leave

### Beschreibung:

Zum Verlassen eines Servers mit gegebener Id. Gilt nur für Mitglieder. Ein Serverbesitzer kann diese nicht verlassen, sondern nur löschen. Liefert eine Erfolgsmeldung zurück.

### Parameter:

- id: String

### Beispiel:

```
> curl -X POST -H "userKey: xxxxx"  
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/leave
```

### Antwort:

```
{  
  "status": "success",  
  "message": "Successfully exited",  
  "data": {}  
}
```

## POST /servers/:id/invites

### Beschreibung:

Erzeugt ein neuen Einladungslink für den Server mit der gegebenen Id. Nur der Serverbesitzer kann Einladungen erstellen. Eine Einladung kann temporal 24 Stunden gültig sein oder für eine gewisse Menge von Personen gelten. Liefert den Einladungslink mit Metainformationen zurück.

### Parameter:

- id: String

### Body:

- type: String (temporal, count)
- max: Integer (optional, nur wenn type = count)

### Beispiel:

```
> curl -X POST -H "userKey: xxxxx" https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/invites  
-d '{ "type": "count", "count": 15 }'
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "id": "5e2ffbd8770dd077d445qs900",  
    "link":  
      "https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/invites/5e2ffbd8770dd077d445qs900",  
    "type": "count",  
    "max": 15,  
    "current": 0,  
    "server": "5e2ffbd8770dd077d03df505"  
  }  
}
```

## POST /servers/:id/invites/:invId

### Beschreibung:

Der eingeloggte Benutzer verwendet diesen Einladungslink, um einem Server beizutreten. Liefert zurück, ob der Beitritt funktioniert hat.

### Parameter:

- id: String
- invId: String

### Body:

- name: String
- password: String

### Beispiel:

```
> curl -X POST -H "userKey: xxxxx"
  https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/invites/5e2ffbd8770dd077d445qs900
  -d '{ "name": "Mr", "password": "Spock" }'
```

### Antwort:

```
{
  "status": "success",
  "message": "Successfully arrived at server",
  "data": {}
}
```

## GET /servers

### Beschreibung:

Liefert eine Liste aller Server zurück, die dir gehören oder bei denen du Mitglied bist.

### Beispiel:

```
> curl -X GET -H "userKey: xxxxx" https://ac.uniks.de/api/servers
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": [
    {
      "id": "5e2ffbd8770dd077d03df505",
      "name": "TestServer"
    }
  ]
}
```

## GET /servers/:id

### Beschreibung:

Liefert explizite Informationen zu einem Server.

### Parameter:

- id: String

### Beispiel:

```
> curl -X GET -H "userKey: xxxxx" https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "id": "5e2ffbd8770dd077d03df505",
    "name": "TestServer",
    "owner": "5e2ffbdabg75dd077d03df505",
    "categories": [ "5e2ffbdabg75dd078d03df600" ],
    "members": [ {"id": "5e2ffbdabg75dd077d03df505", "name": "Spock", "online": true,
      "description": "Spielt gerade AmongUs" } ]
  }
}
```

## GET /servers/:id/invites

### Beschreibung:

Liefert eine Liste aller Einladungen des Servers mit der gegebenen Id zurück. Diese Schnittstelle kann nur von einem Serverbesitzer verwendet werden.

### Beispiel:

```
> curl -X GET -H "userKey: xxxxx" https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/invites
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": [
    {
      "id": "5e2ffbd8770dd077d445qs900",
      "link":
        "https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/invites/5e2ffbd8770dd077d445qs900",
      "type": "count",
      "max": 15,
      "current": 0,
      "server": "5e2ffbd8770dd077d03df505"
    }
  ]
}
```



## PUT /servers/:id

### Beschreibung:

Updated den Namen eines Servers. Liefert den veränderten Server zurück. Diese Schnittstelle kann nur von einem Serverbesitzer verwendet werden.

### Parameter:

- id: String

### Body:

- name: String

### Beispiel:

```
> curl -X PUT -H "userKey: xxxxx" https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505  
-d '{"name": "AndererServerName"}'
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "id": "5e2ffbd8770dd077d03df505",  
    "name": "AndererServerName"  
  }  
}
```

## DELETE /servers/:id

### Beschreibung:

Löscht den Server mit der gegebenen Id, dabei werden allen Kategorien und Kanäle mitgelöscht. Ist nur möglich, wenn dir der Server gehört. Liefert eine minimale Variante des gelöschten Servers zurück.

### Parameter:

- id: String

### Beispiel:

```
> curl -X DELETE -H "userKey: xxxxx" https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "id": "5e2ffbd8770dd077d03df505",  
    "name": "AndererServerName"  
  }  
}
```

## DELETE /servers/:id/invites/:invId

### Beschreibung:

Löscht einen Einladungslink mit der gegebenen Id aus einem Server. Diese Schnittstelle kann nur von einem Serverbesitzer verwendet werden. Liefert die gelöschte Einladung zurück.

### Parameter:

- id: String
- invId: String

### Beispiel:

```
> curl -X DELETE -H "userKey: xxxxx"  
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/invites/5e2ffbd8770dd077d445qs900
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "id": "5e2ffbd8770dd077d445qs900",  
    "link":  
      "https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/invites/5e2ffbd8770dd077d445qs900",  
    "type": "count",  
    "max": 15,  
    "current": 0,  
    "server": "5e2ffbd8770dd077d03df505"  
  }  
}
```

## POST /servers/:id/categories

### Beschreibung:

Erzeugt eine neue Kategorie in dem Server mit der gegebenen Id.

### Parameter:

- id: String

### Body:

- name: String

### Beispiel:

```
> curl -X POST -H "userKey: xxxxx" https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories -d '{"name": "TestKategorie" }'
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "id": "5e2ffbd8770dd077d03df600",
    "name": "TestKategorie",
    "server": "5e2ffbd8770dd077d03df505",
    "channels": []
  }
}
```

## GET /servers/:id/categories

### Beschreibung:

Liefert eine Liste aller Kategorien eines Servers zurück.

### Parameter:

- id: String

### Beispiel:

```
> curl -X GET -H "userKey: xxxxx" https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": [
    {
      "id": "5e2ffbd8770dd077d03df600",
      "name": "TestKategorie",
      "server": "5e2ffbd8770dd077d03df505",
      "channels": []
    }
  ]
}
```

## PUT /servers/:id/categories/:catId

### Beschreibung:

Updated eine Kategorie in dem Server mit der gegebenen Id. Gibt die Kategorie mit den neuen Werten zurück.

### Parameter:

- id: String
- catId: String

### Body:

- name: String

### Beispiel:

```
> curl -X PUT -H "userKey: xxxxx"
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600
-d '{ "name": "TestKategorieGeändert" }'
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "id": "5e2ffbd8770dd077d03df600",
    "name": "TestKategorieGeändert",
    "server": "5e2ffbd8770dd077d03df505",
    "channels": []
  }
}
```

## DELETE /servers/:id/categories/:catId

### Beschreibung:

Löscht eine Kategorie in dem Server mit der gegebenen Id. Gibt die gelöschte Kategorie zurück.

### Parameter:

- id: String
- catId: String

### Beispiel:

```
> curl -X DELETE -H "userKey: xxxxx"  
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "id": "5e2ffbd8770dd077d03df600",  
    "name": "TestKategorieGeändert",  
    "server": "5e2ffbd8770dd077d03df505",  
    "channels": []  
  }  
}
```

## POST /servers/:id/categories/:catId/channels

### Beschreibung:

Erstellt einen neuen Kanal in dem Server/Kategorie mit der entsprechenden Id. Liefert den erzeugten Kanal zurück.

### Parameter:

- id: String
- catId: String

### Body:

- name: String
- type: String (text, audio)
- privileged: boolean
- members: String[] (optional, nur wenn privileged = true)

### Beispiel:

```
> curl -X POST -H "userKey: xxxx"
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels
-d '{ "name": "TestKanal", "type": "text", "privileged": false }'
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "id": "5e2ffbd8770dd077d03dt445",
    "name": "TestKanal",
    "type": "text",
    "privileged": false,
    "category": "5e2ffbd8770dd077d03df600",
    "members": [],
    "audioMembers": []
  }
}
```

## POST /servers/:id/categories/:catId/channels/:chanId/join

### Beschreibung:

Tritt einem Audiokanal bei. War das Beitreten erfolgreich, liefert diese Schnittstelle eine Aufforderung zum Öffnen eines UDP Sockets zurück.

### Parameter:

- id: String
- catId: String
- chanId: String

### Beispiel:

```
> curl -X POST -H "userKey: xxxx"
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels/
5e2ffbd8770dd077d03dt447/join
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "id": "5e2ffbd8770dd077d03dt447"
  }
}
```

## POST /servers/:id/categories/:catId/channels/:chanId/leave

### Beschreibung:

Verlässt einen Audiokanal. Liefert eine Meldung über das Erfolgreiche oder nicht Erfolgreiche Verlassen des Kanales zurück.

### Parameter:

- id: String
- catId: String
- chanId: String

### Beispiel:

```
> curl -X POST -H "userKey: xxxx"
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels/
5e2ffbd8770dd077d03dt447/leave
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "id": "5e2ffbd8770dd077d03dt447"
  }
}
```

## GET /servers/:id/categories/:catId/channels

### Beschreibung:

Liefert eine Liste aller Kanäle einer Kategorie eines Servers zurück.

### Parameter:

- id: String
- catId: String

### Beispiel:

```
> curl -X GET -H "userKey: xxxxx"  
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": [  
    {  
      "id": "5e2ffbd8770dd077d03dt445",  
      "name": "TestKanal",  
      "type": "text",  
      "privileged": false,  
      "category": "5e2ffbd8770dd077d03df600",  
      "members": [],  
      "audioMembers": []  
    }  
  ]  
}
```



## PUT /servers/:id/categories/:catId/channels/:chanId

### Beschreibung:

Upgraded einen Kanal in dem/der Server/Kategorie mit der entsprechenden Id. Liefert den geänderten Kanal zurück.

### Parameter:

- id: String
- catId: String
- chanId: String

### Body:

- name: String
- privileged: boolean
- members: String[] (optional, nur wenn privileged = true)

### Beispiel:

```
> curl -X PUT -H "userKey: xxxxx"
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels/
5e2ffbd8770dd077d03dt445
-d '{ "name": "TestKanalAnders", "type": "text", "privileged": true,
      "members": ["5e2ffbdabg75dd077d03df505"] }'
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "id": "5e2ffbd8770dd077d03dt445",
    "name": "TestKanalAnders",
    "type": "text",
    "privileged": true,
    "category": "5e2ffbd8770dd077d03df600",
    "members": [ "5e2ffbdabg75dd077d03df505" ] ,
    "audioMembers": []
  }
}
```

## DELETE /servers/:id/categories/:catId/channels/:chanId

### Beschreibung:

Löscht einen neuen Kanal aus dem Server/Kategorie mit der entsprechenden Id. Liefert den gelöschten Kanal zurück.

### Parameter:

- id: String
- catId: String
- chanId: String

### Beispiel:

```
> curl -X DELETE -H "userKey: xxxxx"  
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels/  
5e2ffbd8770dd077d03dt445
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "id": "5e2ffbd8770dd077d03dt445",  
    "name": "TestKanalAnders",  
    "type": "text",  
    "privileged": true,  
    "category": "5e2ffbd8770dd077d03df600",  
    "members": [ "5e2ffbdabg75dd077d03df505" ] ,  
    "audioMembers": []  
  }  
}
```

## GET /servers/:id/categories/:catId/channels/:chanId/messages?timestamp=1616935874361

### Beschreibung:

Liefert eine Liste von Chat Nachrichten eines Kanales einer Kategorie eines Servers zurück. Es werden die letzten 50 Nachrichten, die älter als der gegebene *timestamp* sind, geschickt.

### Parameter:

- id: String
- catId: String
- chanId: String

### Query Parameter:

- timestamp: long

### Beispiel:

```
> curl -X GET -H "userKey: xxxxx"  
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels/  
5e2ffbd8770dd077d03dt445/messages?timestamp=1616935874361
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": [  
    {  
      "id": "5e2ffbd8770dd077d03dr458",  
      "channel": "5e2ffbd8770dd077d03dt445",  
      "timestamp": 1616935874361,  
      "from": "Bob",  
      "text": "Hello fellowship!"  
    }  
  ]  
}
```

## PUT /servers/:id/categories/:catId/channels/:chanId/messages/:msgId

### Beschreibung:

Updated den Text einer Chatnachricht mit gegebener Id in einem Channel. Liefert die geänderte Nachricht zurück. Es können nur selbst verfasste Nachrichten geändert werden.

### Parameter:

- id: String
- catId: String
- chanId: String
- msgId: String

### Body:

- text: String

### Beispiel:

```
> curl -X PUT -H "userKey: xxxxx"
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels/
5e2ffbd8770dd077d03dt445/messages/5e2ffbd8770dd077d03dr458
-d { "text": "Hello fellowship and all other people out there!" }
```

### Antwort:

```
{
  "status": "success",
  "message": "",
  "data": {
    "id": "5e2ffbd8770dd077d03dr458",
    "channel": "5e2ffbd8770dd077d03dt445",
    "timestamp": 1616935874361,
    "from": "Bob",
    "text": "Hello fellowship and all other people out there!"
  }
}
```

## DELETE /servers/:id/categories/:catId/channels/:chanId/messages/:msgId

### Beschreibung:

Löscht eine Chatnachricht mit gegebener Id in einem Channel. Liefert die gelöschte Nachricht zurück. Es können nur selbst verfasste Nachrichten gelöscht werden.

### Parameter:

- id: String
- catId: String
- chanId: String
- msgId: String

### Beispiel:

```
> curl -X DELETE -H "userKey: xxxxx"  
https://ac.uniks.de/api/servers/5e2ffbd8770dd077d03df505/categories/5e2ffbd8770dd077d03df600/channels/  
5e2ffbd8770dd077d03dt445/messages/5e2ffbd8770dd077d03dr458
```

### Antwort:

```
{  
  "status": "success",  
  "message": "",  
  "data": {  
    "id": "5e2ffbd8770dd077d03dr458",  
    "channel": "5e2ffbd8770dd077d03dt445",  
    "timestamp": 1616935874361,  
    "from": "Bob",  
    "text": "Hello fellowship and all other people out there!"  
  }  
}
```

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

Die Beispiele für die Asynchrone Kommunikation bestehen lediglich aus beispielhaften JSON Nachrichten und der dazugehörigen URL.

## /system

### Beschreibung:

Auf diesem Channel werden System Nachrichten versendet. Zum Beispiel, ein Benutzer hat sich gerade angemeldet.

Auf diesen Channel können keine Nachrichten gesendet werden, mit Ausnahme der *noop* Operation.

**Beispiel:** `wss://ac.uniks.de/ws/system`

### Nachrichtenaufbau (Antworten):

```
{
  "action": "<action>",
  "data": {
    ...
  }
}
```

Aktionen: "userJoined", "userLeft", "userDescriptionChanged"

Das Daten Feld variiert zwischen den Aktionen.

## /system?serverId=5e2ffbd8770dd077d03df505

### Beschreibung:

Auf diesem Channel werden System Nachrichten eines bestimmten Servers versendet. Zum Beispiel, ein Benutzer ist jetzt Online oder neue Kategorien/Kanäle wurden angelegt/gelöscht.

Auf diesen Channel können keine Nachrichten gesendet werden, mit Ausnahme der *noop* Operation.

**Beispiel:** `wss://ac.uniks.de/ws/system?serverId=5e2ffbd8770dd077d03df505`

### Nachrichtenaufbau (Antworten):

```
{
  "action": "<action>",
  "data": {
    ...
  }
}
```

Aktionen: "userJoined", "userLeft", "userArrived", "userExited", "serverUpdated", "serverDeleted", "categoryCreated", "categoryUpdated", "categoryDeleted", "channelCreated", "channelUpdated", "channelDeleted", "messageUpdated", "messageDeleted", "inviteExpired", "audioJoined", "audioLeft"

Das Daten Feld variiert zwischen den Aktionen.

## **/chat?user=MyUserName**

### **Beschreibung:**

Auf diesem Kanal können private Nachrichten an einzelne Nutzer gesendet werden.

### **Query Parameter:**

- user: String

### **Befehle:**

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

**Beispiel:** `wss://ac.uniks.de/ws/chat?user=TestUser`

```
{  
  "channel": "private",  
  "to": "Bob",  
  "message": "Hallo Welt nur an dich!"  
}
```

### **Nachrichtenaufbau (Antworten):**

```
{  
  "channel": "private",  
  "timestamp": 1614938273193,  
  "message": "Hello World!",  
  "from": "Karli",  
  "to": "Bob"  
}
```

## **/chat?user=MyUserName&serverId=5e2ffbd8770dd077d03df505**

### **Beschreibung:**

Auf diesem Kanal können Nachrichten an Textkanälen eines Servers gesendet werden.

### **Query Parameter:**

- user: String
- serverId: String

### **Befehle:**

- Nachricht an einen bestimmten Kanal
  - channel: String
  - message: String

**Beispiel:** `wss://ac.uniks.de/ws/chat?user=TestUser&serverId=5e2ffbd8770dd077d03df505`

```
{  
  "channel": "5e2ffbd8770dd077d03dt445",  
  "message": "Hallo Welt an alle hier!"  
}
```

### **Nachrichtenaufbau (Antworten):**

```
{  
  "id": "5e2ffbd8770dd077d03dr458",  
  "channel": "5e2ffbd8770dd077d03dt445",  
  "timestamp": 1614938273193,  
  "from": "Bob",  
  "text": "Hallo Welt an alle hier!",  
}
```



# Audiostream UDP ([cranberry.uniks.de:33100](http://cranberry.uniks.de:33100))

Für das Übermitteln von Audiodaten wird das UDP Protokoll verwendet. Dabei müssen Datagram Pakete zu exakt 1279 Bytes verschickt werden. Die ersten 255 Bytes sind für Metadaten im Json Format vorgesehen. Für den Fall, dass der resultierende Json String nicht 255 Bytes lang ist, müssen die restlichen Bytes mit „nullen“ aufgefüllt werden.

Die restlichen 1024 Byte sind für Audiodatenpakete reserviert.

Die Metadaten müssen wie folgt aufgebaut werden:

```
{  
  "channel": "5e2ffbd8770dd077d03dt445",  
  "name": "Bob",  
}
```

Für die Audiodatenpakete müssen folgende Spezifikationen eingehalten werden:

```
BITRATE      = 48 kHz  
SAMPLE_SIZE  = 2 Byte  
CHANNELS     = 1  
DATA_SIGNING = true  
BYTE_ORDER   = little-endian
```