

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>.
- 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"  
    }  
  ]  
}
```

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 /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"
  }
}
```

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"} ]
  }
}
```

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": "MyAwesomeCategory",  
      "server": "5e2ffbd8770dd077d03df505",  
      "channels": []  
    }  
  ]  
}
```

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": "MyAwesomeChannel",  
      "type": "text",  
      "privileged": false,  
      "category": "5e2ffbd8770dd077d03df600",  
      "members": []  
    }  
  ]  
}
```

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!"  
    }  
  ]  
}
```


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"

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"

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!",  
}
```