

KUID

Eine KUID ist eine bestimmte Nummer, die zur Referenzierung von Trainz-Inhalten dient

Eine KUID ist eine bestimmte Nummer, die zur Referenzierung von Trainz-Inhalten dient. Jedes Objekt (zB. Lok, Waggon, Fahrgestell, Gleis, Bodentextur oder auch Layout) hat eine KUID, die aus 2-3 Zahlenblöcken besteht.

Ein Objekt kann hierbei auf ein anderes Objekt über dessen KUID verweisen, so kann beispielsweise in der config.txt der Lok über die KUID auf das entsprechende Interior verwiesen werden.

Jeder der sich bei Auran registriert hat, erhält automatisch eine USER-ID, die man, wenn man sich bei Auran anmeldet, unter dem Link "Your USER ID" findet. Diese ist sehr wichtig für die KUID!

Anhand einer KUID kann nämlich Content und der Autor identifiziert und durch die [Datenbank](#) verwaltet werden.

Dabei gibt es zwei KUID-Versionen (KUID und KUID2), die man benutzen kann:

KUID sieht so aus: <KUID:x:y>

KUID2 sieht so aus: <KUID2:x:y:z>

x = USER-ID

y = Nummer des Contents (frei wählbar)

z = Versionsnummer (je höher, umso "aktueller" der Content, außerdem wird von Trainz nur die aktuelle Version genutzt)

Wer also Inhalte hochladen möchte, sollte stets immer auf die KUID achten und die Benutzung von fremden USER-IDs vermeiden!

KUID:

Standardmäßig wird das KUID-Format genutzt. Hierbei besteht der erste Ziffernblock aus der Auran-Benutzer-ID (jedes Planet Auran - Konto hat eine ID), beispielsweise 12345. Den zweiten Teil der KUID stellt

die ID des Objektes dar, beispielsweise 6789.

Eine KUID kann also folgendermaßen aussehen:

<kuid:12345:6789>

KUID2:

Das KUID2-Format ist dann sinnvoll, wenn überarbeitete Versionen eines Objektes herausgegeben werden sollen.

Das KUID2-Format erweitert das KUID-Format um eine Versionsnummer. Diese gibt die Version des jeweiligen Objektes wieder und wird als 3. Segment an die KUID angehängt, so lautet eine KUID2 beispielsweise:

<kuid2:12345:6789:3>

Ein Objekt im KUID-Format (z.B. <kuid:10101:99999>) entspricht im KUID2-Format <kuid2:10101:99999:0>. Die Version 1 stellt bereits die erste Überarbeitung/Revision dar.

Hinweis:

Bei UTC und TRS2004 sollte man beim ersten Start von Trainz eine USER-ID eingeben, damit Trainz Strecken und Sessions automatisch unter der eigene USER-ID speichert. Bei TRS2006 und höher sollte man im CMP die Einstellungen bearbeiten, dort den Benutzernamen und Passwort eingeben und mit dem Internet verbunden sein, damit Trainz automatisch den selbst erstellten Content unter der eigenen USER-ID speichert.

Tut man dies nicht, benutzt Trainz einen Platzhalter und benutzt als USER-ID die -2. Wer nun Content mit der USER-ID -2 veröffentlicht, wird eine böse Überraschung erleben. Denn wenn Benutzer, die ihre Benutzerdaten im CMP eingeben haben, nun den Content importieren, wird die USER-ID automatisch gegen die des aktuellen Benutzers ausgetauscht. Allerdings wird die USER-ID nicht in der KUID-Table angerührt, so dass Content fehlende Abhängigkeiten anzeigt.

Warum wird nun der Platzhalter benutzt?

Der gesamte Content wird durch die [Datenbank](#) verwaltet und diese benötigt (mindestens) einen eindeutigen Schlüssel also die KUID. Einen leeren Wert (null) als KUID verweigert die [Datenbank](#), denn ein Schlüssel muss existieren, damit die [Datenbank](#) den Content verwalten kann, also muss eine KUID existieren. Da die [Datenbank](#) nur die KUID als Schlüssel akzeptiert, sind andere Informationen nicht relevant wie z.B. der

Name des Objekts. So ist es möglich Contents zu benutzen, die den gleichen Namen, aber eine unterschiedliche KUID haben.

Mögliche Fehlerquellen:

Beispiel 1:

Baum 1 <KUID:123:1>

Auto 1 <KUID:123:1>

Beispiel 2:

Baum 1 <KUID:123:1>

Baum 1 <KUID:124:2>

Bei Beispiel 1 handelt es sich hierbei um den gleichen Content wegen der KUID. Obwohl es zwei unterschiedliche Objekte gibt, akzeptiert die [Datenbank](#) nur ein Objekt, das andere Objekt wird sie nicht bearbeiten, da es ja schon vorhanden ist, weil die [Datenbank](#) Content an der einzigartigen KUID erkennt, aber nicht an anderen Eigenschaften wie z.B. Name oder Größe der einzelnen Dateien. Bei Beispiel 2 handelt es sich hierbei um zwei unterschiedliche Objekte wegen der KUID. Es kann sein, dass die Objekte gleich sind, aber anhand der KUID behandelt die [Datenbank](#) diese als jeweils zwei unterschiedliche Objekte.

Wer also Content erstellt oder vorhandenen Content bearbeitet und diesen veröffentlichen möchte, sollte seine eigene USER-ID für die KUID benutzen! Wenn man ein Reskin für ein vorhandenes Objekt erstellt hat, sollte man die original KUID durch die eigene ersetzen. Auch wenn man Strecken baut und man seine Benutzerdaten nicht im CMP eingegeben hat, sollte man spätestens vor der Veröffentlichung noch mal die KUID überprüfen