# SQL Server Datenbank Design -> Kroki ER Diagramm

https://kroki.io/examples.html#erd

[crayon-662a9cac644e2428340104/]

# SQL Query um Tabellen und Spalten auszulesen inkl. PK und FK

[crayon-662a9cac648cc956641605/]

#### Host oder Webhost mit Autofac

[crayon-662a9cac64a39379716531/]

# Entity Framework EF — Tabellen Models generieren lassen

Alt + T -> "NU" eingeben [crayon-662a9cac64bcc644963402/]

# Windows Powershell Dienst einrichten

[crayon-662a9cac64d02583619870/]

## SQL Server Statistik IO und Time

[crayon-662a9cac64e4c077412878/]

## Nuxt 3 mit Keycloak autorisieren

In diesem Beitrag will ich eine Schritt für Schritt Anleitung geben, wie man Nuxt 3 mit Keycloak autorisiert.

Die Thematik um Autorisierung und Authentifizierung bedarf einer soliden Kenntnis in der Thematik. Ich empfehle daher unbedingt sich die Zeit zu nehmen und das folgende Video anzuschauen und sicher zu gehen, dass alles verstanden wurde:

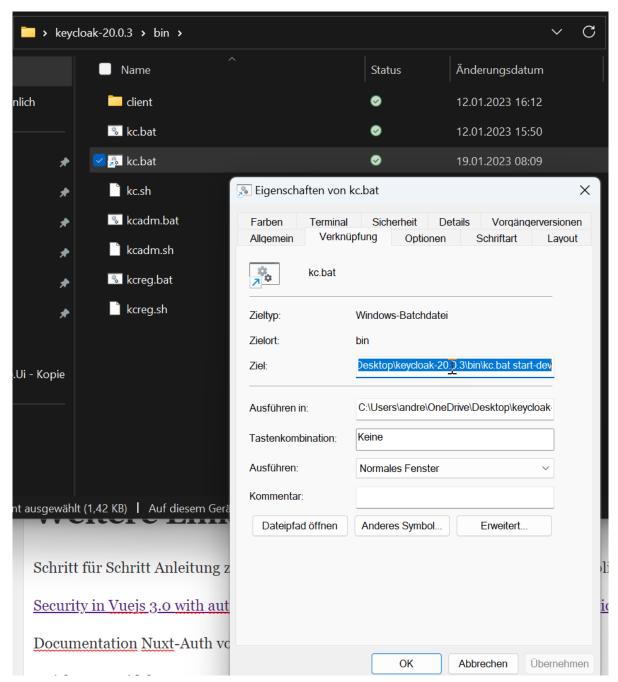
Zum Ende füge ich weitere Lohnenswerte Links hinzu.

### 1. Keycloak installieren

1. Keycloak herunterladen: downloads - Keycloak



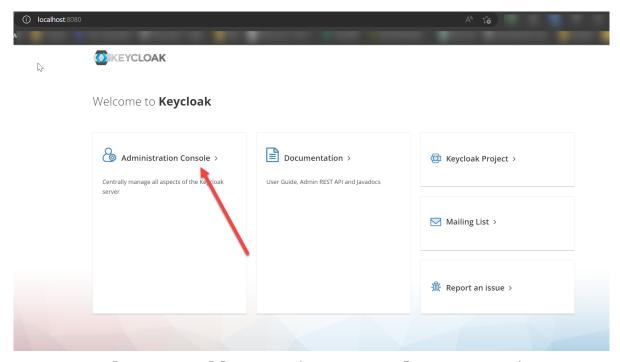
2. Zip-Datei entpacken und in das **bin** Verzeichnis wechseln. Dort eine Verknüpfung vom **kc.bat** erstellen, dann auf Eigenschaften und beim Ziel **start-dev** anhängen



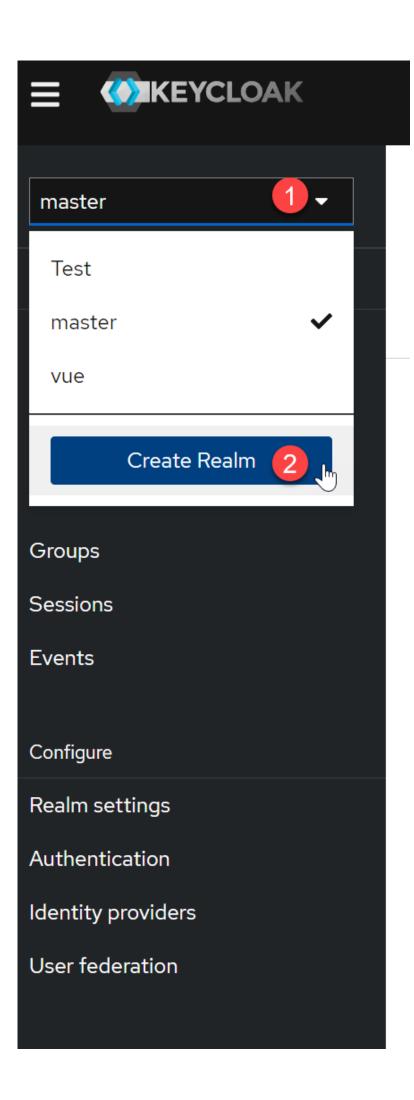
3. Verknüpfung starten und keycloak ist unter der Adresse http://localhost:8080 erreichbar. Dort einen neuen Admin User anlegen

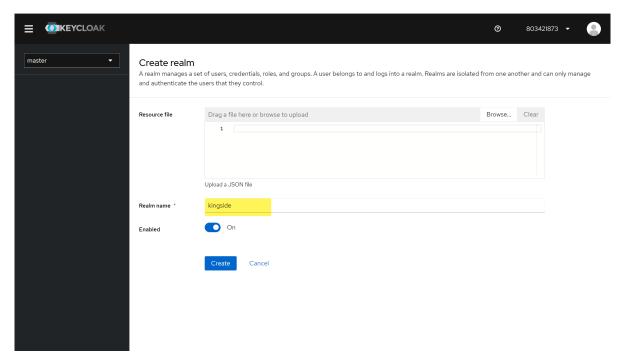
#### 2. Keycloak einrichten

1. Keycloak Administarion console aufrufen über http://localhost:8080

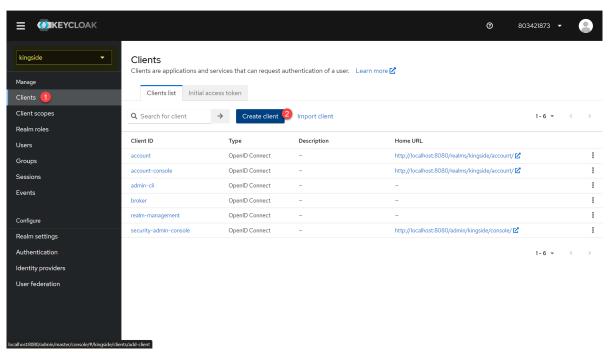


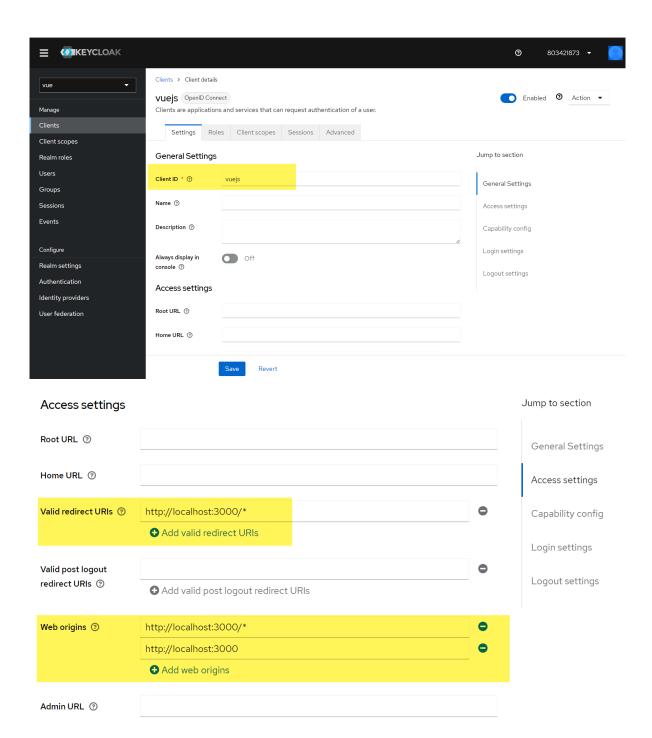
2. Neuen realm erstellen und nur Realm name eintragen. Wichtig! Keine Sonderzeichen benutzen





3. Jetzt in dem angelegten realm einen neuen Client erstellen



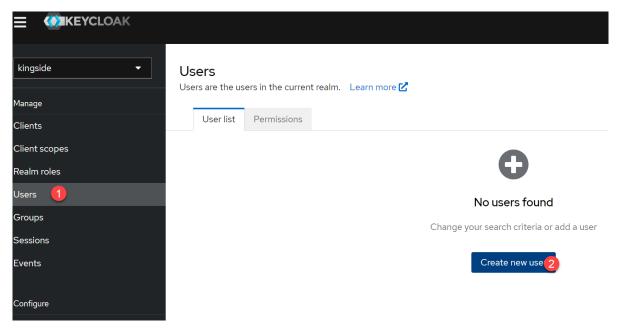


#### Capability config Client authentication ③ Off Authorization ③ Off Authentication flow Standard flow ③ Direct access grants 🕙 Implicit flow ③ Service accounts roles 3 OAuth 2.0 Device Authorization Grant $\odot$ OIDC CIBA Grant 🗿 Login settings Login theme ③ Choose... Consent required ② Off Display client on Off screen ② Logout settings Front channel logout On 3 Front-channel logout URL ③ Backchannel logout URL ③ **Backchannel logout** On session required ③ Backchannel logout Off revoke offline sessions 3

Revert

Save

#### 4. User erstellen. Der **Username** reicht



#### Create user

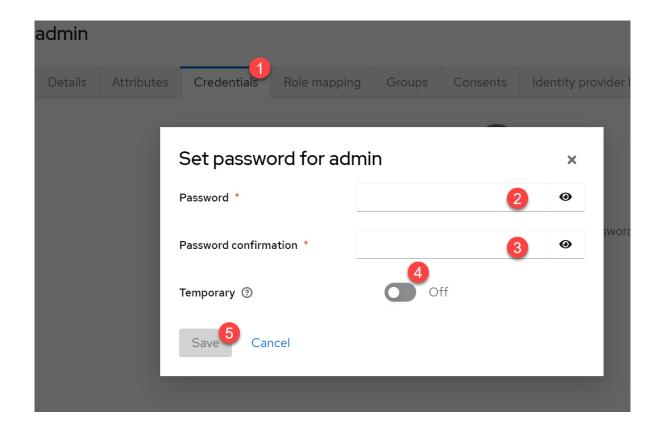
Username *	admin
Email	
Email verified ②	Off
First name	
Last name	
Required user actions	Select action
(2)	Scient detion
Groups ②	Join Groups
	Create Cancel

5. Zu dem User wechseln und folgende Sachen um **Credentials** ergänzen

#### Users > User details

#### admin

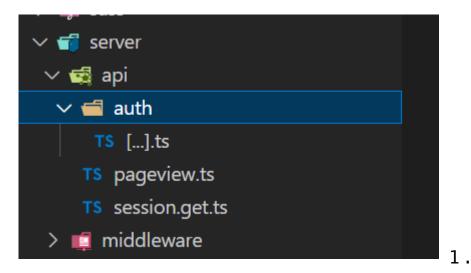
Details	Attributes	Credential	s Role mappir
ID *	f	6c242e6-cab	5-4323-8d76-96
Created at *	1,	/30/2023, 10:	:50:24 AM
Username *	а	dmin	
Email			
Email verifie	d ③	On	



#### 3. nuxt 3 App aufsetzen

Installiertes nodeJs und pnpm sind Voraussetzung

- 1. Visual studio Code starten und das prime-vue starter repository clonen:
  - https://github.com/sfxcode/nuxt3-primevue-starter.git
- 2. Verzeichnis auswählen wo die repo hin soll und mit öffnen bestätigen
- 3. Wir wechseln zum Terminal und installieren sidebase nuxt-auth. Zur Zeit des Schreibens existiert auch eine 0.4.0 alpha Version. Diese Version verursacht aber Fehler mit Keycloak.:
  - [crayon-662a9cac64fc1904624443/]
- 4. nun in der nuxt.config.ts die modules ergänzen und nuxt mitteilen, dass Autorisierung auf jeder Seite gelten soll:
  - [crayon-662a9cac64fc4995021299/]
- 5. Jetzt muss unter server 2 Dateien erstellt werden:



server\auth\[...].ts

[crayon-662a9cac64fc6080420622/]

und 2. server\api\session.get.ts

[crayon-662a9cac64fc7973339447/]

6. Eine weitere Datei muss angelegt werden für die Umgebungsvariablen: **.env** 

[crayon-662a9cac64fc8937952025/]

Lasst euch ein Secret irgendwo generieren. Dieser String ist nur für Demozwecke!!!

Wichtig ist, dass die lokale Adresse mit 127.0.0.1 angegeben wird. localhost oder 0.0.0.0 kann nicht aufgelöst werden. In den Url ist der real name enthalten.

#### Weitere Links

Schritt für Schritt Anleitung zur lokalen Einrichtung von Keycloak und einer Vue Applikation:

Security in Vuejs 3.0 with authentication and authorization by KeyCloak Part 1 | by Nicolas Barlatier | Dev Genius

Documentation Nuxt-Auth von Sidebase:

Ouick Start · sidebase

Nuxt-Auth wurde von Next-Auth portiert. Fehlende Dokumentation kann hier nachgelesen werden:

Introduction | NextAuth.js (next-auth.js.org)

Ich nutze das prime-vue starter Paket, welches nuxt 3 inkludiert hat. Ich kann das Paket jeden nur empfehlen!

sfxcode/nuxt3-primevue-starter: Build your VUE.js App with Nuxt3 . First Class PrimeVUE support. Formkit Validation included. (github.com)

Für OAuth und OpenId Connectt gibt es Debugger. Der Author ist Nathon aus dem YouTube Video weiter oben.

OAuth 2.0 debugger (oauthdebugger.com)

OpenID Connect debugger (oidcdebugger.com)

Keycloak Dokumentation:

Documentation - Keycloak

# Nuxt3 mit Vuetify 3 — Teil 3: Entwickeln in Vue / Typescript

Typescript muss in VS Code installiert werden. -g steht für globally (Alle Projekte)
[crayon-662a9cac651a0434577525/]
Typescript Version prüfen
[crayon-662a9cac651a3605513758/]

#### **Enums**

[crayon-662a9cac651a4407667401/]

Objekte werden mit reactive<type>({...}) reactiviert und primitive Datentypen mit ref<string>(...)

[crayon-662a9cac651a6484390697/]

Mixins werden in eine Vue Instanz hinzugefügt. So als würde man 2 Vue Instanzen verschmelzen. Composables sind dem ähnlich. In Mixins wird alles importiert, bei Composables muss jede Variable, Methode, etc. einzeln importiert werden.

Guter Artikel dazu:

What is a Vue.js Composable? — Vue.js Tutorials (vueschool.io)

# Nuxt3 mit Vuetify 3 — Teil 2: Layout

Wenn man durch eine App navigiert, dann ist der Seitenaufbau überall identisch. Das Menü, die Kopfzeile und Fusszeile sind auf jeder Seite identisch. Lediglich der Inhalt ändert sich.

In Nuxt 3 werden Layouts im "layouts"-Ordner erstellt und anschließend in die app.vue hinzugefügt. Einzelne Seiten werden im "pages"-Ordner erstellt.

Nuxt 3 routet die pages anhand des Namens. Nur die page "index.vue" wird zum root Knoten geroutet.

Also erstellen wir ein Verzeichnis "layouts" und ein Verzeichnis "pages".

In das "layouts" Verzeichnis, eine Datei erstellen
"default.vue".

und folgenden Inhalt kopieren:
[crayon-662a9cac6531f819351218/]

Dort, wo <slot /> steht, kommt der Inhalt der einzelnen Pages hinein.

Jetzt in das Verzeichnis "pages"

die Datei index.vue mit folgenden Inhalt [crayon-662a9cac65322242046272/] und die Datei "pageOne.vue" mit folgenden Inhalt erstellen [crayon-662a9cac65323045646751/]

In der app.vue müssen wir nun sagen, dass wir nicht mehr die Nuxt Welcome page sehen wollen, sondern unser Layout:

[crayon-662a9cac65325026308401/]

# Nuxt3 mit Vuetify 3 — Teil 1: Installation

Wenn man aus der Microsoft Welt kommt und Webseiten mit JQuery bearbeitet hat, dann sind UI Framworks wie Vue eine wahre Veränderung. In meiner ASP WebApp habe ich Vue2 über CDN eingebunden und Vue anstatt JQuery genutzt. Vue funktioniert aber am besten mit NodeJs. Der Versuch NodeJs zu erlernen verlief aber immer in Frust. Weil es keinen roten Faden gibt, wie etwas umgesetzt werden kann. Es gibt keinen besten Weg. Es gibt sehr viele neue Begriffe und Definitionen, die Stück um Stück erlernt werden müssen. Es ist definitiv ein anderes lernen wie eine erprobte Sprache wie C# oder Java.

Deshalb ist hier **mein** roter Faden. Ziel soll am eine App sein, die folgende Punkte mit sich bringt:

- 1. Nuxt 3 als Wrapper über Vue3. Nuxt 3 schafft eine Ordnerstruktur und macht die imports z.B. überflüssig.
- 2. Vuetify 3 Beta. Zwar ist es noch in Beta Phase, aber es wird das beste UI Framework (Meiner Meinung nach)
- 3. Typescript: Die App soll in Typescript geschrieben sein
- 4. Pinia wird unser Statemanagement anstatt Vuex
- 5. Übersetzungen werden über i18n realisiert
- 6. Daten werden über externe .NET 6 Minimal Api mit Swagger abgerufen
- 7. Authorisierung erfolgt über eine ASP .NET 6 Proxy Anwendung

#### **Installation**

#### Voraussetzung

NodeJS, Visual Studio Code, Volar Extension. Aktuelle Links sind hier zu finden:

Nuxt 3 - Quick Start (nuxtjs.org)

In Visual Studio über "Strg + Shift + ö" das Terminal öffnen

Neue App erstellen:

[crayon-662a9cac654c1800460572/]

Die App wird nun im Unterverzeichnis "nuxt-app" erstellt.

Daher müssen wir in das neue Verzeichnis navigieren

[crayon-662a9cac654c4669905960/]

packages nachinstallieren:

[crayon-662a9cac654c5635122952/]

#### **Vuetify installieren**

#### Referenzen:

The Easy Way to add Vuetify Vue.js 3 Material Components to Your Nuxt 3 / Vue.js 3 Application — YouTube

[Solved] vuetify icon not showing — Local Coder (Solution 5)

vuetify - npm (npmjs.com)

Unter "Version" die neuste Version raussuchen

[crayon-662a9cac654c6818275861/]

Sass muss ebenfalls für Vuetify installiert werden

[crayon-662a9cac654c7805648361/]

Pinia als StateManager

[crayon-662a9cac654c8085141751/]

[crayon-662a9cac654c9257880608/]

und die mdi Icons für Vuetify

[crayon-662a9cac654ca901196840/]

Plugins werden in Nuxt im plugins Verzeichnis erstellt. Also muss im Root ein neues Verzeichnis "plugins" erstellt

In das plugins Verzeichnis wird eine neue Datei "vuetify.ts" erstellt und folgendes kopiert:

[crayon-662a9cac654cb614618163/]

Jetzt muss noch die nuxt.config.ts angepasst werden und das css und der transpiler müssen mit vuetify verheiratet werden. Daher kann der folgende Teil einfach copy/pasted werden. Der Vite Teil wird bei mir als Fehler angezeigt. kompilieren geht trotzdem. Wem das stört, kann den vite Knoten auskommentieren [crayon-662a9cac654cc395282306/]

Jetzt sollte die App starten können:

In der Beta 1 läuft dauert der Start über 1 Minute. Ich hoffe das wird noch gefixt

[crayon-662a9cac654ce220372514/]

Weiter geht's mit Teil 2 Layout:

https://devandy.de/?p=890&preview=true