

# Konzept für Service-Manager

Christopher Mogler

April 2020

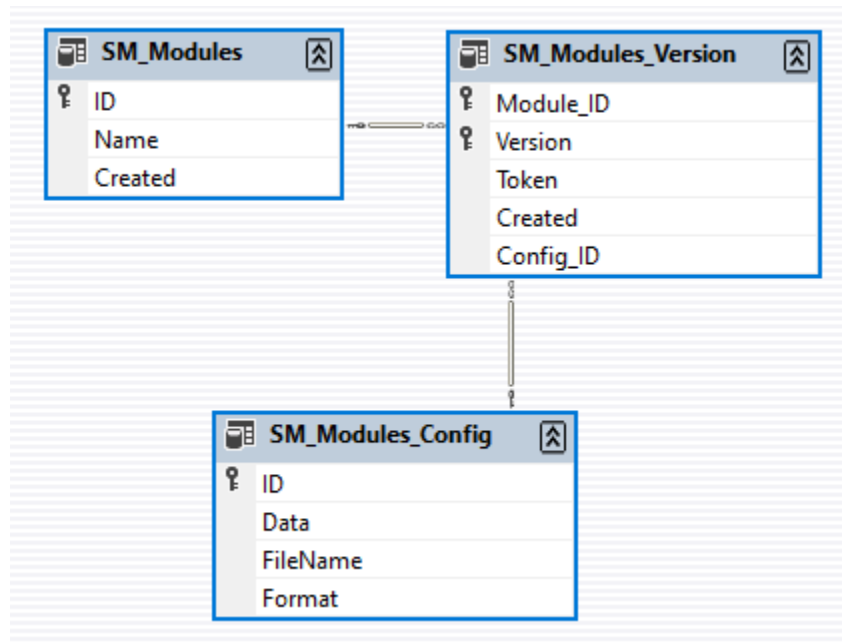
## Inhalt

<b>1</b>	<b>SM.Database</b>	<b>2</b>
1.1	Module . . . . .	2
1.1.1	SM_Modules . . . . .	2
1.1.2	SM_Modules_Versions . . . . .	2
1.1.3	SM_Modules_Config . . . . .	2
1.2	Customer . . . . .	3
1.2.1	SM_Customers . . . . .	3
1.2.2	SM_Customers_Modules . . . . .	3
<b>2</b>	<b>SM.API</b>	<b>3</b>
2.1	Informationsabruf . . . . .	3
2.1.1	Datenaufbau . . . . .	3
2.2	Module . . . . .	4
2.3	Informationsänderung . . . . .	4
<b>3</b>	<b>SM.Service</b>	<b>5</b>
3.1	Ablauf . . . . .	5
<b>4</b>	<b>SM.UI</b>	<b>6</b>
4.1	Oberfläche . . . . .	6

# 1 SM.Database

## 1.1 Module

Die Module sind in einer Datenbank-Tabelle hinterlegt. Die einzelnen Versionen sind in einer extra Tabelle hinterlegt.



### 1.1.1 SM\_Modules

Hier sind alle Module, die über SM Installiert werden können, hinterlegt.

- **ID** ist GUID für jedes Modul.
- **Name** ist ein String/Varchar der den Name des Modul beinhaltet.
- **Created** ist ein DateTime/TimeStamp der wiedergibt wann dieser Satz erstellt wurde.

### 1.1.2 SM\_Modules\_Versions

Alle Versionen pro Moduls, sind hier hinterlegt.

- **Module\_ID** hier wird die ID vom Modul hinterlegt, um die Version mit dem Modul zu verknüpfen. Es ist eine "1-zu-n"-Verknüpfung.
- **Version** die aktuelle Version als String/Varchar. Diese ist der Primärschlüssel der Tabelle.
- **Token** zum abgleichen für den Service, ob eine Änderung durchgeführt wurde.
- **Created** wann wurde diese Version hinzugefügt.
- **Config\_ID** für die Version dazugehörige Konfigurationsdatei. Diese ist mit der SM\_Modules\_Config-Tabelle verknüpft.

### 1.1.3 SM\_Modules\_Config

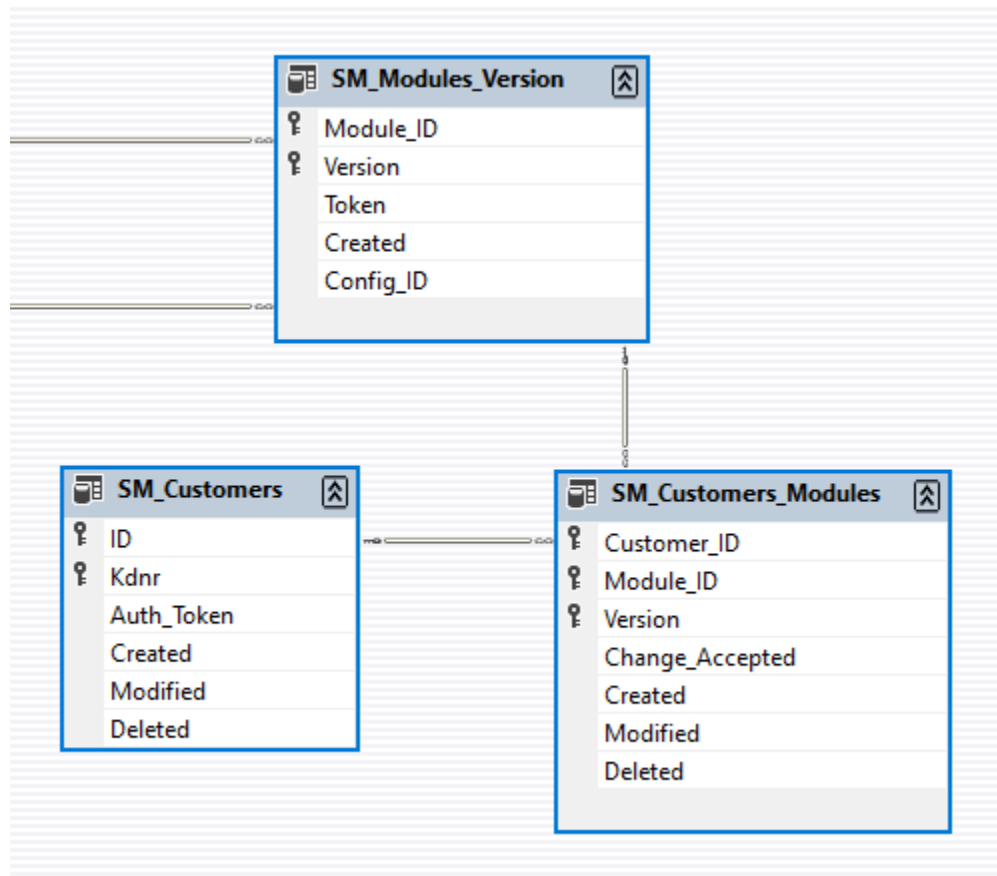
Hier wird die Konfigurationsdatei für jede Version gespeichert.

- **ID** die GUID der Konfigurationsdatei.
- **Data** der Inhalt als BLOB der Konfigurationsdatei.

- **FileName** der Dateiname als String, wie die die Konfigurationsdatei heißen muss.
- **Format** der Format der Konfigurationsdatei, ob es eine JSON oder XML, etc. Format hat.

## 1.2 Customer

Hier wird definiert welcher Kunde hat zugriff auf welche Module. Und auch wird der Authentifizierungstoken hinterlegt, womit sich der Dienst bei der API authentifizieren kann.



### 1.2.1 SM\_Customers

TODO

### 1.2.2 SM\_Customers\_Modules

TODO

## 2 SM.API

### 2.1 Informationsabruf

Um die Daten abzurufen muss sich die Schnittstelle authentifizieren. Dies erfolgt durch einen Token der für jeden Kunden einmalig ist.

#### 2.1.1 Datenaufbau

Die Daten zum Abrufen der Informationen sind im JSON-Format. Bsp. wie die JSON-Datei aufgebaut ist:

```

{
  "version": "1.0",
  "changed": "2020-04-02 12:12:12.222",
  "token": "50ad41624c25e493aa1dc7f4ab32bdc5a3b0b78ecc35b539936e3fea7c565af7",
  "modules": [
    {
      "id": "048e4a4e-ab6a-482e-9f34-8821325527f6",
      "name": "MailInterface",
      "version": "12.0.1",
      "token": "ad6af7308eab03163a9cbef49f22636efdb0f884620d777a29a8e8fc7634f967"
    },
  ]
}

```

Erklärung:

- **version:** die aktuelle Version von Service-Manager.
- **changed:** letzte Änderung vom aktuellen Datensatz
- **token:** wird bei Änderungen verändert, damit die Schnittstelle schneller auf Änderungen prüfen kann. Ohne alle Module einzeln zu Prüfen.
- **modules:** alle Module die beim Kunden installiert sein muss.
  - **id:** ID vom Modul, wird verwendet um die Datei später herunterzuladen, über die API.
  - **name:** Name des Moduls
  - **version:** Aktuelle Version die Bereitgestellt wird.
  - **token:** Prüfungstoken um die installierte Version abzugleichen auf Änderungen.

## 2.2 Module

Die Anwendungen werden in einem bestimmten Verzeichnis aufbewahrt. Jede Version einer Anwendung wird hinterlegt, als ZIP mit der Version als Dateiname. Bsp.: `/Anwendungen/MailInterface/1.14.1.zip`  
 Der Pfad kann deshalb Dynamisch hergestellt werden, aus den Daten der Tabellen.

## 2.3 Informationsänderung

## 3 SM.Service

### 3.1 Ablauf

Es wird eine Verbindung zur REST-API (siehe *SM.API*) hergestellt. Die Informationen werden als JSON-Format übergeben. Die Daten werden serialisiert in Objekte, damit die Verarbeitung vereinfacht wird.

## 4 SM.UI

### 4.1 Oberfläche

TODO