

REST API  
 DAS WICHTIGSTE AUF  
 EINEN BLICK

# SMAPONE API INFORMATIONEN

## Grundlegendes:

Deatillierte API-Dokumentation: <https://platform.smapone.com/backend/swagger/ui/index>

Basisroute: <https://platform.smapone.com/Backend/intern>

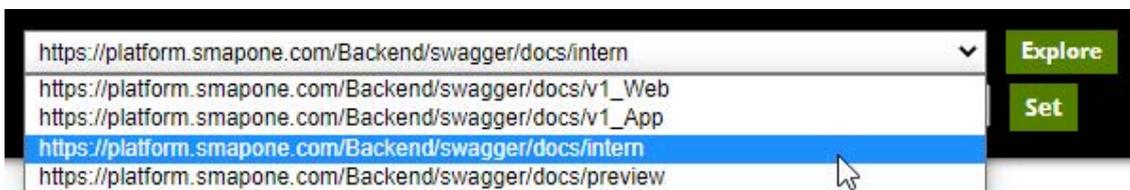
## Authentifikation:

### 1. Möglichkeit:

Basic Authentication, der User kann leer bleiben, als Kennwort muss der REST-API Token des Creatoren Account angegeben werden, der REST-API Token kann unter <https://platform.smapone.com/portal/Creator/Profile/ManageApiToken> im Creatoren Account generiert bzw. wieder ausgelesen werden

### 2. Möglichkeit:

über URL Query Parameter, an jede Router an die REST-API kann der Parameter „accessToken“ angehangen werden. Als Wert wird der REST-API Token gesetzt. Eine URL sieht dann Beispielsweise so aus: <https://platform.smapone.com/Backend/.../Data?accessToken=abcdefghijklmnopqrstuvwxyz01234567890>



## Abruf Datensätze:

GET /Smaps/{smapId}/Data  
GET /Smaps/{smapId}/Versions/{majorVersion}/Data  
GET /Smaps/{smapId}/Versions/{version}/Data  
GET /Smaps/{smapId}/Versions/{version}/Data/{recordId}.{format}

Der Abruf der Datensätze, hier zutreffend auf die ersten 3 Varianten liefert jeweils eine Liste von Datensätzen. Der letztgenannte Abruf bezieht sich auf einen einzelnen Datensatz und ermöglicht neben dem Abruf als JSON auch die weiteren Formate PDF, Word etc.

## Erstellung von Aufgaben:

**Router zur Funktion:** POST /Smaps/{smapId}/Versions/{version}/Data

### Parameter:

- smapId > ID der Smap zu welcher die Aufgaben erstellt werden soll
- version > Version der Smap zu welcher die Aufgabe erstellt werden soll, die Version gibt dabei maßgeblich Aufbau der Smap an
- Body > siehe nächste Seite

### Beispiel Body:

```
{
  „userEmail“: „michael.rath@smapone.com“,
  „title“: „mein Titel“,
  „data“: {
    „Textbox_Name“: „Beispiel Name“,
    „Textbox_Anschrift“: „Beispiel Anschrift“,
    „Number_Hausnummer“: 24
  }
}
```

#### userEmail

Muss mit der E-Mailadresse eines Nutzer befüllt werden. Die Smap muss ebenfalls an den Nutzer zugeordnet sein.

#### title

Der Titel kann frei gewählt werden, darf max. 25 Zeichen aufnehmen.

#### data

Beim „data“ Attribut handelt es sich um ein Objekt welches den Inhalt der Aufgabe enthält.

Um Bausteine über eine Aufgabe zu befüllen, müssen diese als Key-Value Pairs angegeben werden. Als Key wird die Baustein-Id verwendet welche der Creator für diesen festgelegt hat. Der Value richtet sich nach dem Typen des Bausteins. Typen können zum Beispiel einfache Datentypen wie Texte, Zahlen oder Datumswerte sein.

Gültige Datumsformate (je nach Einstellung des Bausteins gültig):

einfaches Datum: „2021-10-19“

einfache Uhrzeit: „19:11:00“

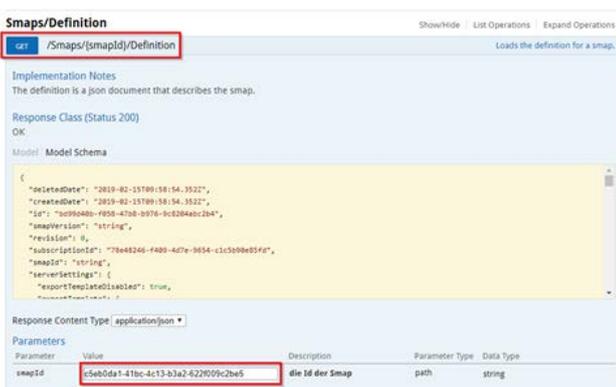
Datum mit Uhrzeit: „2021-10-19T19:11:00“

Zeitpunkt: „2021-10-19T19:11:37.7902909+02:00“

Darüber hinaus sind noch viele weitere komplexe Bausteine vorhanden welche befüllt werden können. Das Format für die Befüllung eines Bausteins entspricht immer auch dem Ausgabeformat in einem manuell erfassten Datensatz.

### Aktualisierung einer Datensatzauswahl:

1. Baustein „Datensatzauswahl“ im Designer an die gewünschte Position platzieren
2. Excelliste mit den gewünschten Spalten ggf. erzeugen und mit mindestens einer Datenzeile in den Baustein laden
3. unter dem API Aufruf „GET /Smaps/{smapId}/Definition“ kann anhand der SmapId der Aufbau der smapOne App angesehen werden, hier werden Sie zum Datensatzauswahl Baustein die Id der Datasource finden, die dort festgehaltene Datasource Id ist gleichbleibend, solange der Baustein durch den Creator nicht entfernt wird



4. Für jede Aktualisierung der Datenliste in der Datasource sind einige Details notwendig, diese können via „GET /DataSource/{dataSourceId}“ abgerufen werden.
5. Via „PUT /DataSource/{dataSourceId}/Versions/{dataSourceVersion}/Definition/Values“ laden Sie Ihre neue Datenliste hoch, das Schema muss zu den in der Datasource festgehaltenen Spalten passen, es wird dabei immer die komplette Datenliste aktualisiert – ein partielles Update ist aktuell nicht vorgesehen.
6. Um die aktualisierte Liste an alle mobilen Endgeräte zu verteilen nutzen Sie zum Abschluss „PUT /Smaps/{smapId}/Versions/Current/DataSources/Update“, die notwendigen SmapIds können Sie aus dem Abruf aus Punkt 6 entnehmen – Attribut „usedInSmaps“, hier ist ebenso noch der Parameter „updateEditVersion=true“ zu setzen, so dass die aktualisierte Liste für den Creator im Designer ebenfalls erscheint.

Die Schritte 1 - 3 sind dabei für die einmalige Einrichtung notwendig.

Die Schritte 4 - 6 sind der Teil welcher durch einen Automatismus abgedeckt werden sollte.