# Schnittstellen

## Angebotene Schnittstellen

### Produkte

Die Anwendung ‚Referenzimplementierung REST‘ bietet anderen Fachanwendungen Schnittstellen zur Suche (Anlage, Änderung u. Löschen) von Produktdaten.

#### NST\_Produkt\_lesen\_mit\_ID

|  |  |
| --- | --- |
| **Kurzbeschreibung** | Eine andere Fachanwendung fragt anhand des Eingabeparameter ‚Produkt-ID‘ ein dadurch eindeutig bestimmtes Produkt an.  Das unter der Produkt-ID gefundene Produkt wird an das anfragende System übermittelt. |
| **Version** | 1.0 |
| **Synchron/Asynchron** | Synchron, Lesend |
| **Input-/Output-Parameter** | * In: Produkt-ID (UID) - Pflicht, nicht-leer * Out: Produkt (Siehe hierzu 2.1.1.) – nicht-leer |
| **Fachliche Fehlermeldungen** | * Fehlermeldung „Produkt nicht gefunden“:  Eine Fehlermeldung, falls mit den als Input Parameter übergebenen Produkt-ID kein Produkt gefunden wird. |
| **Implementierung** | AwfProduktSuchen |

#### NST\_Produkte\_lesen\_mit\_Name

|  |  |
| --- | --- |
| **Kurzbeschreibung** | Eine andere Fachanwendung fragt anhand des Eingabeparameter ‚Produkt-Name‘ eine Liste von Produkten mit diesem Namen an. Da der Name des Produktes nicht einmalig sein muss, kann es mehrere Treffer geben. Wenn der Eingabeparameter leer ist, werden alle aktuell vorhandenen Produkte ausgegeben.  Eine Liste der gefundenen Produkte wird an das anfragende System übermittelt. |
| **Version** | 1.0 |
| **Synchron/Asynchron** | Synchron, Lesend |
| **Input-/Output-Parameter** | In: Produkt-Name (Text) – Optional, leer erlaubt Out: Liste von Produkten (Siehe hierzu 2.1.1.) - leer möglich |
| **Fachliche Fehlermeldungen** | -/- |
| **Implementierung** | AwfProduktSuchen |

## Benutzte Schnittstellen

### Produkte

Die Anwendung ‚Referenzimplementierung REST‘ bietet anderen Fachanwendungen Schnittstellen zur Suche (Anlage, Änderung u. Löschen) von Produktdaten.

#### NST\_Produkt\_suchen\_mit\_ID

|  |  |
| --- | --- |
| **Aufrufstelle** | AFU\_Produktdaten\_aktualisieren |
| **Kurzbeschreibung** | Über das eindeutige Merkmal ‚Produkt-ID‘ wird ein Produkt angefragt. |
| **Version** | 1.0 |
| **Anbietendes System** | Anwendungssystem ‚Referenzimplementierung REST‘ |
| **Synchron/Asynchron** | Synchron, Lesend |
| **Input-/Output-Parameter** | * In: Produkt-ID (UID) - Pflicht, nicht-leer * Out: Produkt (Siehe hierzu 2.1.1.) – nicht-leer |
| **Fachliche Fehlermeldungen** | * Fehlermeldung „Produkt nicht gefunden“:  Eine Fehlermeldung, falls mit den als Input Parameter übergebenen Produkt-ID kein Produkt gefunden wird. |
| **Implementierung** | AwfProduktAnzeigen |

#### NST\_Produkt\_suchen\_mit\_Name

|  |  |
| --- | --- |
| **Aufrufstelle** | AFU\_Produktdaten\_aktualisieren |
| **Kurzbeschreibung** | Mithilfe des Produkt-Attributs ‚Produkt-Name‘ werden Produkte mit diesem Namen angefragt. Ergebnis ist eine (evtl. leere) Liste mit Produkten, die diesen Namen haben. |
| **Version** | 1.0 |
| **Anbietendes System** | Anwendungssystem ‚Referenzimplementierung REST‘ |
| **Synchron/Asynchron** | Synchron, Lesend |
| **Input-/Output-Parameter** | * In: Produkt-Name (Text) – leer möglich * Out: Liste von Produkten (Siehe hierzu 2.1.1.) – leer möglich |
| **Fachliche Fehlermeldungen** | -/- |
| **Implementierung** | AwfProduktAnzeigen |

# Fachliches Datenmodell

## Schnittstellenentitätstypen

### NSE\_ProduktModel

###### Primäre Attribute

| **Attribut** | **Datentyp** | **Pflicht?** | **Beschreibung** |
| --- | --- | --- | --- |
| ATT\_ID | DTY\_UID | Ja | Eindeutige Identifikations-nummer.  Einmalig, nur intern. |
| ATT\_Name | DTY\_Text | Ja | Name des Produkts |
| ATT\_Beschreibung | DTY\_Text | Nein | Beschreibungstext zum Produkt |

# Transformation zu OpenAPI-Spezifikation

* Angebotene Schnittstellen = Service:   
  Erstellung der OpenAPI-Spezifikation und Veröffentlichung zum Gebrauch durch Anwendungen, die diese Schnittstellen als Client verwenden.   
  Es entsteht eine yaml-Datei, die die angebotene Schnittstelle beschreibt - bzw. bei großen fachlich-thematischen Unterschieden: mehrere Dateien, fachlich-thematisch getrennt.
* Benutzte Schnittstellen = Client:   
  Verwendung der von der den Service anbietenden Anwendung erstellten und veröffentlichten OpenAPI-Spezifikation.   
  Eine Anwendung kann Schnittstellen von mehreren anderen Anwendungen verwenden und jede anbietende Anwendung liefert (mindestens) eine yaml-Datei, die die von ihr angebotene Schnittstelle beschreibt.
* PATH u. HTTP-Methode (Pflicht):   
  ergibt sich aus dem Namen und der Funktion/Beschreibung der Schnittstelle.   
  Gemäß REST-Konzept: 3.1. Aufbau der Ressourcen-URIs u. 3.2. Verwendung von HTTP-Methoden
* tags (Service: Optional / Client: Pflicht):   
  Gruppierung von Operationen (Client: -> Name der API-Klasse)
* description (Optional):   
  Kurzbeschreibung (-> JavaDoc)
* operationId (Pflicht):   
  aus Schnittstellenname (-> Methodenname)
* GET: parameters / POST, PUT, PATCH: requestBody:   
  (Referenz auf) Input-Parameter/-Objekte
* responses.’200’.content.application/json.schema:   
  (Referenz auf) Output-Objekt
* components.schemas :  
  Definition der Input- u. Output-Objekte mit ihren Attributen (->model.<DTO>)