**Lastenheft**

**‚FLOG-IT WEB‘**

Anbindung eines webbasierten1 Artikelversands mit Datenbank2

für **Sören Jacobshagen**

**SabRoDan  
support@SabRoDan.de**  
Thalbergstraße 13  
99086 Erfurt

Ansprechpartner :

Sabine Lück, Roland Peyerl, Daniel Wandrowec

Inhalt

[1. Zielbestimmung 4](#_Toc359661789)

[2. Produkteinsatz 4](#_Toc359661790)

[3. Produktübersicht 4](#_Toc359661791)

[4. Produktfunktionen 5](#_Toc359661792)

[4.1 Funktion: Nutzer Registrieren4 5](#_Toc359661793)

[4.2 Funktion: Nutzer anmelden 5](#_Toc359661794)

[4.3 Funktion: Nutzer abmelden 5](#_Toc359661795)

[4.4 Funktion: Rechnung ändern 5](#_Toc359661796)

[4.5 Funktion: Rechnung erstellen 6](#_Toc359661797)

[4.6 Funktion: Kontaktdaten abrufen 6](#_Toc359661798)

[4.7 Funktion: Kontaktdaten filtern 6](#_Toc359661799)

[4.8 Funktion: Kontakt suchen 6](#_Toc359661800)

[4.9 Funktion: Kontakt anlegen 7](#_Toc359661801)

[4.10 Funktion: Kontakt bearbeiten 7](#_Toc359661802)

[4.11 Funktion: Kontakt löschen 7](#_Toc359661803)

[4.12 Funktion: Angebot Liste abrufen 8](#_Toc359661804)

[4.13 Funktion: Angebot Details abrufen 8](#_Toc359661805)

[4.14 Funktion: Angebot anlegen 8](#_Toc359661806)

[4.15 Funktion: Angebot bearbeiten 8](#_Toc359661807)

[4.16 Funktion: Status des Angebotes ändern 10](#_Toc359661808)

[4.17 Funktion: Angebot löschen 10](#_Toc359661809)

[4.18 Funktion: Produkt Liste abrufen 10](#_Toc359661810)

[4.19 Funktion: Produkt anlegen 10](#_Toc359661811)

[4.20 Funktion: Produkt Details abrufen 10](#_Toc359661812)

[4.21 Funktion: Produkt bearbeiten 11](#_Toc359661813)

[4.22 Funktion: Produkt löschen 11](#_Toc359661814)

[4.23 Funktion: Produktsuche 11](#_Toc359661815)

[4.24 Funktion: Rechnung suchen 11](#_Toc359661816)

[5. Produktdaten 13](#_Toc359661817)

[5.1 Artikel 13](#_Toc359661818)

[5.7 Angebot 13](#_Toc359661819)

[5.8 Rechnung 14](#_Toc359661820)

[5.9 Nutzer 14](#_Toc359661821)

[6. Produktleistungen 15](#_Toc359661822)

[7. Qualitätskriterien 16](#_Toc359661823)

[8. Glossar 16](#_Toc359661824)

# 1. Zielbestimmung

Die Software3 soll den webbasierten Vertrieb von eigenen Artikeln ermöglichen. Das Sortiment soll sich je nach Verkäufer unterscheiden können. Die Benutzung soll im Wesentlichen die Verwaltung und Nutzung von Internetauktionsplattformen verbessern.

# 2. Produkteinsatz

Das Produkt dient der Verwaltung von Auktionen von eigenen Büchern, CDs, DVDs und Ähnlichem. Die Verwaltung und Analyse der Angebotenen Artikel soll dabei vereinfacht werden. Momentan erfolgt die Organisation mit einer Tabellenverwaltung, durch die Software soll der Vorgang der Organisation beschleunigt werden, ebenso können nun mehrere Nutzer gleichzeitig ein unterschiedliches Sortiment haben.

3. Produktübersicht

****

# 4. Produktfunktionen

## 4.1 Funktion: Nutzer Registrieren4

|  |  |
| --- | --- |
| Funktionsbeschreibung | Ein Nutzer kann sich, sofern noch nicht geschehen, beim System registrieren, um sich anmelden zu können |
| Input | * Name, neues Passwort + Passwort bestätigen |
| Output | * Rückmeldung über Erfolg und Misserfolg der Registrierung |
| Vorbereitungen | * keine |
| Nachbedingungen | * Nutzer ist registriert |
| Fehlerfälle | * Nutzername schon vorhanden * Passwort falsch bestätigt * Zugriff auf DB gescheitert |

## 4.2 Funktion: Nutzer anmelden

|  |  |
| --- | --- |
| Funktionsbeschreibung | Ein Nutzer kann sich beim System anmelden, um zu einem erweiterten Funktionsumfang Zugriff zu erlangen |
| Input | * Name + Passwort |
| Output | * Rückmeldung Erfolg / Misserfolg der Anmeldung |
| Vorbereitungen | * User muss registriert sein |
| Nachbedingungen | * Nutzer ist angemeldet |
| Fehlerfälle | * User ist nicht registriert * Zugriff auf DB gescheitert |

## 4.3 Funktion: Nutzer abmelden

|  |  |
| --- | --- |
| Funktionsbeschreibung | Ein Nutzer kann sich vom System abmelden, um unerlaubte und unerwünschte Zugriffe durch andere zu verhindern |
| Input | * Abmelden bestätigt |
| Output | * Rückmeldung Erfolg / Misserfolg der Abmeldung |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Nutzer ist abgemeldet |
| Fehlerfälle |  |

## 4.4 Funktion: Rechnung ändern

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann den Bezahlstatus und/oder ein Lieferdatum in einer Rechnung auf "offen" oder "bezahlt" bzw. „offen“ (kein Datum eingetragen) oder „verschickt“ (Datum) |
| Input | * Datum oder bezahlt |
| Output | * Anzeige ob Änderung erfolgreich |
| Vorbereitungen | * Nutzer ist angemeldet * Rechnung ist selektiert |
| Nachbedingungen | * Status wurde geändert |
| Fehlerfälle | * Bei ungültiger Eingabe --> Neueingabe erforderlich |

## 4.5 Funktion: Rechnung erstellen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Ein Nutzer kann sich zu einem bestimmten Angebot eine Rechnung erstellen lassen |
| Input | * Alle Daten aus bereits vorhandenem Daten (*AngebotID5, Gesamtpreis (aus Einzelpreis \* Anzahl + Lieferkosten) , Kontakt des Käufers, Kaufdatum* ) |
| Output | * Ausgabe der Rechnung bei Erfolg |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Bezahlt-Status der Rechnung auf „offen“, Versand-Status auf „offen“, Interne ID zugewiesen |
| Fehlerfälle | * Zugriff auf DB gescheitert |

## 4.6 Funktion: Kontaktdaten abrufen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann, nach dem er sich am System angemeldet hat, sein Kontaktbuch einsehen, welches eine Liste der von Ihm angelegten Kontakte beinhaltet. |
| Input |  |
| Output | * Alle Kontakte die der angemeldete Nutzer angelegt hat |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Anzeigen des Bestellungszählers, nach Überprüfung der Rechnungen |
| Fehlerfälle |  |

## 4.7 Funktion: Kontaktdaten filtern

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit beim Ansehen seines Kontaktbuches die Daten nach bestimmten Kriterien filtern zu lassen. |
| Input | * Auswählen der Art der Filterung   + Alphabetische Anordnung nach Name/Anschrift etc.   + Nach Anzahl Bestellungen |
| Output | * Alle Kontakte die der angemeldete Nutzer angelegt hat |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * keine |

## 4.8 Funktion: Kontakt suchen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit Kontakte direkt suchen zu lassen. |
| Input | * Name, ggf. Anschrift |
| Output | * Alle Kontakte, auf die die jeweiligen Suchkriterien zutreffen |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * Kontakt existiert nicht in der DB ->eine Meldung wird angezeigt das die Suchanfrage 0 Treffer ergab * Rechnung nicht verfügbar in der DB ->sollte die Rechnung von dem Kunden nicht auffindbar sein wird die Anzahl der Rechnungen als "0" angezeigt |

## 4.9 Funktion: Kontakt anlegen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit neue Kontakte in sein Kontaktbuch hinzuzufügen. |
| Input | * Pflicht Eingabe von Vorname, Nachname, Anschrift, * Optionale Angaben: email, Telefon, Mobil, Titel |
| Output | * Anzeige des neu Angelegten Kontaktes |
| Vorbereitungen | * Nutzer ist angemeldet, |
| Nachbedingungen | * Bestellungszähler wird auf 0 gesetzt, Kontakt wird in die DB eingetragen wenn es zu keinem Fehler in der Kontrolle kommt |
| Fehlerfälle | * Kontakt existiert bereits in der DB ->Eingabe Felder Vorname + Nachname werden überprüft in Kombination mit der Anschrift, sollte der Kontakt schon existieren bekommt der Nutzer einen Hinweis * Kontakt enthält Fehlerhafte Daten -> mit Hilfe von try+catch6 Kontrolle und Vermeidung von Eingaben mit fehlerhaften Daten |

## 4.10 Funktion: Kontakt bearbeiten

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit Kontaktdaten zu bearbeiten. |
| Input | * wie beim „Anlegen des Kontaktes“ |
| Output | * Anzeige des geänderten Kontaktes |
| Vorbereitungen | * Nutzer ist angemeldet, Kontakt Daten wurden geladen und werden angezeigt |
| Nachbedingungen | * Kontrolle auf fehlerhafte Eingaben (Leerfelder und falsche Formateingabe) |
| Fehlerfälle | * Siehe Kontakt anlegen |

## 4.11 Funktion: Kontakt löschen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit Kontakte in sein Kontaktbuch zu löschen. |
| Input | * Auswählen des zu löschenden Kontaktes durch ID bzw. Name |
| Output | * Meldung das Löschung erfolgreich |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Der Rechnungszähler (Bestellungen) ist auf 0 |
| Fehlerfälle | * Kontakt existiert nicht in der DB -> der Nutzer bekommt einen Hinweis das der von ihm gewählte Kontakt nicht mehr in der DB existiert * Fehler beim Löschen des Kontakt in der DB->Hinweis, Kontakt wird wieder hergestellt * Kontakt besitzt Verknüpfungen zu Rechnungen-> keine Löschung möglich -->Benutzer wird darauf hingewiesen |

## 

## 4.12 Funktion: Angebot Liste abrufen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann nach dem er sich an das System angemeldet hat, die Liste seiner angelegten Angebote einsehen. |
| Input | * Keine |
| Output | * Alle Angebote die der angemeldete Nutzer angelegt hat |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * Nutzer besitzt keine Angebote-> in dem Fall wird eine Leere Liste angezeigt |

## 4.13 Funktion: Angebot Details abrufen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann von der vorhergehenden Liste aus ein Angebot auswählen von dem er aus die Details einsehen möchte. |
| Input | * Auswahl des Angebots (ID Weitergabe durch Button klick) |
| Output | * Angebote Details |
| Vorbereitungen | * Nutzer ist angemeldet, ID wurde mit gesendet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * ID ist falsch/nicht mit übergeben ->Fehlermeldung |

## 4.14 Funktion: Angebot anlegen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit durch Button klick in der Angebots Liste ein neues Angebot anzulegen auf einer extra Seite.  Es wird hierbei 1 Eintrag in die DB gemacht: Angebot anlegen. |
| Input | * Angebot   + Pflicht Eingabe von Startpreis, buynowprice, ID des zu verkaufenden Produkts vom Auktionshaus, URL   + Optionale Angaben: Enddatum des Angebots |
| Output | * Anzeige des neu angelegten Angebots wenn erfolgreich auf der Angebots Detail Seite * Bei Fehler --> Rückmeldung mit betreffender Zeile + Neueingabe |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Startdatum des Angebots auf das Erstellungsdatum des Angebots setzen * (Verknüpfung zum angemeldeten Nutzer vermerken durch FK) //Sören arbeitet vorerst allein * Status des Angebots wird beim Erstellen immer auf offen gesetzt |
| Fehlerfälle | * Angebot konnte nicht erstellt werden in der DB ->Eingabe war fehlerhaft, der Nutzer erhält einen Hinweis * Angebot enthält fehlerhafte Daten -> mit Hilfe von Try catch Kontrolle und Vermeidung von fehlerhaften Eingaben |

## 4.15 Funktion: Angebot bearbeiten

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer hat die Möglichkeit Angebote zu bearbeiten. |
| Input | * wie beim „Anlegen des Angebots“ |
| Output | * Anzeige des geänderten Angebots |
| Vorbereitungen | * Nutzer ist angemeldet, Angebots Daten wurden geladen und werden angezeigt |
| Nachbedingungen | * Kontrolle auf fehlerhafte Neueingaben (Leerfelder und falsche Formateingabe) |
| Fehlerfälle | * Siehe Angebot anlegen |

## 4.16 Funktion: Status des Angebotes ändern

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann den Zustand eines Angebots von „offen“ auf „abgelaufen“ oder „bestellt" setzen |
| Input | * Statusänderung (nach „Abgelaufen“/ „Gekauft“) |
| Output |  |
| Vorbereitungen | * Nutzer ist angemeldet * Angebot ist selektiert |
| Nachbedingungen | * Status wurde geändert |
| Fehlerfälle | * Keine |

## 4.17 Funktion: Angebot löschen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann Angebote wieder löschen auf der Angebots Detail Seite. |
| Input | * Angebots ID |
| Output | * Rückmeldung ob Löschen erfolgreich |
| Vorbereitungen | * Nutzer ist angemeldet, existierendes Angebot zum löschen ausgewählt |
| Nachbedingungen | * Kontrolle ob es Verknüpfungen zu Rechnungen gibt, wenn ja, löschen nicht möglich |
| Fehlerfälle | * Angebot existiert nicht in der DB ->Fehlermeldung Löschen ist in diesem Fall nicht möglich * Angebot ist gekoppelt an eine Rechnung -> das Angebot kann in diesem Fall nicht gelöscht werden da erst die Rechnung aus dem System genommen werden muss |

## 4.18 Funktion: Produkt Liste abrufen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann nach dem er sich an das System angemeldet hat, die Liste seiner angelegten Produkte einsehen. |
| Input | * Keine |
| Output | * Alle Produkte die der angemeldete Nutzer angelegt hat |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * Nutzer besitzt keine Produkte -> in dem Fall wird eine Leere Liste angezeigt |

## 4.19 Funktion: Produkt anlegen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann ein neues Produkt in das System einpflegen.  Hierzu werden je nach Produktart Schablonen zur Eingabe benutzt |
| Input | * Typ des Artikels (siehe erbende Klassen von IArtikel) * Typzugehörige Daten (siehe Datenmodell) |
| Output | * Anzeige des angelegten Produktes |
| Vorbereitungen | * Nutzer ist angemeldet |
| Nachbedingungen | * Produktname einzigartig |
| Fehlerfälle | * Produkt existiert bereits --> Nutzer bekommt Hinweis, Neueingabe |

## 4.20 Funktion: Produkt Details abrufen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Der Nutzer kann von der vorhergehenden Liste aus ein Artikel auswählen von dem er aus die Details einsehen möchte. |
| Input | * Auswahl des Artikel (ID Weitergabe durch Button klick) |
| Output | * Artikel Details |
| Vorbereitungen | * Nutzer ist angemeldet, ID wurde mit gesendet |
| Nachbedingungen | * Keine |
| Fehlerfälle | * ID ist falsch/nicht mit übergeben ->Fehlermeldung |

## 4.21 Funktion: Produkt bearbeiten

|  |  |
| --- | --- |
| Funktionsbeschreibung | Nutzer kann ein bestehendes Produkt bearbeiten, erreichen durch Button klick auf der Produkt Detail Seite. |
| Input | * Produktname oder ID |
| Output | * Produktdaten des eingegebenen Produkts |
| Vorbereitungen | * Nutzer ist angemeldet * Produkt existiert |
| Nachbedingungen | * Überprüfung ob geänderte Produktdaten gültig |
| Fehlerfälle | * Produktname/ID nicht vorhanden --> Hinweis, Neueingabe * Produktdaten nicht gültig --> Hinweis, Daten betreffende Neueingabe |

## 4.22 Funktion: Produkt löschen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Nutzer kann ein bestehendes Produkt löschen auf der Produkt Detail Seite. |
| Input | * Produktname oder ID |
| Output |  |
| Vorbereitungen | * Nutzer ist angemeldet * Produkt existiert * existieren Angebote mit dem Produkt wird es zusammen mit dem Angebot gelöscht |
| Nachbedingungen | * Überprüfen, ob Produkt existiert |
| Fehlerfälle | * Produkt existiert nicht --> Hinweis, Neueingabe * existieren Rechnungen mit dem Produkt ist ein Löschen nicht möglich |

## 4.23 Funktion: Produktsuche

|  |  |
| --- | --- |
| Funktionsbeschreibung | Nutzer kann ein bestehendes Produkt suchen und sich die Daten anzeigen lassen. |
| Input | * Produktname oder ID oder ISBN/EAN |
| Output | * Produktdaten des Produkts, auf das die eingegebenen Daten zutreffen |
| Vorbereitungen | * Nutzer ist angemeldet * Produkt existiert |
| Nachbedingungen | * Überprüfen, ob Produkt existiert |
| Fehlerfälle | * Produkt existiert nicht --> Hinweis, Neueingabe |

## 4.24 Funktion: Rechnung suchen

|  |  |
| --- | --- |
| Funktionsbeschreibung | Nutzer kann eine bestehendes Rechnung suchen und sich die Daten anzeigen lassen. |
| Input | * ID und/oder isShipped und/oder isPaid und/oder date |
| Output | * Rechnung auf die die eingegebenen Daten zutrifft |
| Vorbereitungen | * Nutzer ist angemeldet * Rechnung wurde erstellt |
| Nachbedingungen |  |
| Fehlerfälle | * Rechnung wurde nicht gefunden --> Hinweis, Neueingabe |

# 5. Produktdaten

## 5.1 Artikel

|  |  |  |
| --- | --- | --- |
| Typ | Name | Beschreibung |
| Int7 | ID | Interne ID des Produkts (automatisch generiert) |
| String8 | Name/Title | Name des Produkts |
| String\* | Description | Beschreibung des Artikels |
| String | Place | Lagerort des Artikels |
| String\* | Photo | Pfadangabe wo das Produktbild gelagert ist |
| String\* | EAN/ISBN |  |
| String | externID | Kaufplatz Bestellnummer (von Booklocker, ebay, etc) |
| String | type | Buch/Spiel/CD, etc… |

## 5.7 Angebot

|  |  |  |
| --- | --- | --- |
| Typ | Name | Beschreibung |
| Int | ID | Interne ID |
| Float | StartSaleprice | Startpreis |
| Float | BuyNowPrice | Preis bei Sofortkauf |
| Date | Start | Datum wann das Angebot erstellt wurde |
| Date\* | End | Sollte kein Enddatum gegeben sein, handelt es sich um ein Dauerangebot, ansonsten um eine Aktion (Sonderangebot) |
| Artikel | Product | Angebotenes Produkt (Fremdschlüssel auf ProduktID) |
| String | auctionshouse | z.B. ebay,amazon, etc. |
| String | URL | URL11 zum Angebot beim Auktionshaus |
| Enum12 | Status | Abgelaufen  Offen  Gekauft |
| Int | creatorID | Ersteller ID |

## 5.8 Rechnung

|  |  |  |
| --- | --- | --- |
| Typ | Name | Beschreibung |
| Int | ID | Interne ID |
| Kontakt | Buyer | Käufer des Angebots |
| Int | Angebot\_ID | Auf welches Angebot bezieht sich die Rechnung |
| Float | total | Gesamtpreis |
| Date | BoughtAt | Datum vom Kauf |
| Date | ShippedAt | Wenn eingetragen: Verschickt  NULL13: Offen |
| Bool14 | IsPaid | True: Bezahlt  False: offen |
| Date | PaidAt | Wann wurde bezahlt |
| Float | shippingcost | Lieferkosten |

## 5.9 Nutzer

|  |  |  |
| --- | --- | --- |
| Typ | Name | Beschreibung |
| Int | ID | Interne ID |
| string | userName | Benutzername |
| string | pw | Passwort |

# 6. Produktleistungen

/LL 10/Nutzer: Jeder Nutzer hat ein eigene Erkennungs-ID. Daten werden dadurch an die Nutzer gebunden.

/LL 20/Akkumulation: Bei fehlerhafter Eingabe erhält Nutzer eine Liste aller eingegebenen Fehler.

/LL 30/Toleranz: Bei fehlerhaften Eingaben werden die Daten nicht aus den Feldern gelöscht, so dass der Nutzer die Korrektur vornehmen kann, ohne Eingaben zu wiederholen.

/LL 40/Datensicherheit: Es soll abgesichert sein, dass Nutzer nicht auf Daten Einblick haben die von anderen angelegt wurden.

/LL 50/Verfügbarkeit: Es muss sichergestellt sein, dass der Server immer erreichbar ist.

/LL 60/Archivierung: Dokumente wie Rechnungen werden permanent im System gesichert.

# 7. Qualitätskriterien

Stufe 1= sehr wichtig ... Stufe 4=unwichtig

Kriterium Wertung

Erweiterbarkeit 2

Backups 3

Portierbarkeit 4

Effizienz 2

Skalierbarkeit 3

Benutzbarkeit 2

Datenschutz 3

Änderbarkeit 2

Zuverlässigkeit 1

Funktionalität 1

# 8. Glossar

**webbasiert1:** Ein- und Ausgaben im GUI werden über das Internet an das/von dem Programm übertragen. Dazu ist es unerheblich wo der Nutzer sich befindet.

**Datenbank2:** Elektronisches System, in dem große Bestände an Daten zentral gespeichert sind

**Software3:** Gesamtheit des Systems. Umfasst die Programmlogik, die Präsentation und die Datenbank

**Registrieren4:** Speichern der Nutzerdaten im System für späteren Abruf und Zuordnung der Daten zu einem Nutzer

**ID**5: Eindeutige Identifikationsnummer die von der Datenbank verwendet wird

**try+catch6:** Fehlerbehandlung durch die ein Vorgang durchgeführt wird, aber im Fehlerfall das Programm mit einer Meldung reagiert statt abzustürzen

**Int7: Datentyp für Ganzzahlwerte**

**string8: Datentyp für Zeichenketten**

**date9: Datentyp für Datumsangaben**

**float10: Datentyp für Fließkommazahlen**

**URL11: Unified Ressource Locator ("Einheitlicher Quellenanzeiger")**

**Enum12: Aufzählungsdatentyp bei dem Zahlenfolgen Namen zugeordnet werden**

**NULL13: Wert, der das Fehlen eines Wertes anzeigt**

**bool14: Datentyp, der nur 2 Werte ("wahr" und "falsch") annehmen kann**