



WEBWARE 2.0



WW SVC

Beschreibung der Servicefunktionen

Rev 2.0

INHALTSVERZEICHNIS

1 Glossar.....	5
1.1 Allgemeine Begriffe.....	5
1.2 Zugriffsarten für WEBSERVICES.....	6
2 Allgemeines zu Servicefunktionen.....	7
2.1 Namenskonventionen für Objekte, Funktionen und Parameter.....	7
2.2 Hinweise zu freien Selektionsausdrücken.....	7
2.3 Hinweise zur Debug-Unterstützung.....	7
2.4 Standardparameter.....	9
2.5 Rückgabewerte von Servicefunktionen.....	10
2.6 Standard-Rückgabe von GET-Funktionen.....	10
3 Referenz Servicefunktionen.....	12
3.1 ARTIKEL.GET.....	13
3.2 ARTIKEL.PUT.....	16
3.3 ARTIKEL.INSERT.....	17
3.4 ARTIKEL.DELETE.....	18
3.5 ADRESSE.GET.....	19
3.6 ADRESSE.PUT.....	21
3.7 ADRESSE.INSERT.....	22
3.8 ADRESSE.DELETE.....	23
3.9 BELEG.GET.....	24
3.10 BELEG.PUT.....	26
3.11 BELEG.INSERT.....	27
3.12 BELEG.DELETE.....	29
3.13 BELPOS.GET.....	30
3.14 BELPOS.PUT.....	31
3.15 BELPOS.INSERT.....	32
3.16 IDB<IDBID>.GET.....	33
3.17 IDB<IDBID>.PUT.....	35
3.18 IDB<IDBID>.INSERT.....	36
3.19 IDB<IDBID>.DELETE.....	37
3.20 DBK<DBKNR>.GET.....	38

3.21 DBK<DBKNR>.PUT.....	40
3.22 DBK<DBKNR>.INSERT.....	41
3.23 DBK<DBKNR>.DELETE.....	42
3.24 DBP<DBPNR>.GET.....	43
3.25 DBP<DBPNR>.PUT.....	44
3.26 DBP<DBPNR>.INSERT.....	45
3.27 PROJEKT.GET.....	46
3.28 PROJEKT.PUT.....	48
3.29 PROJEKT.INSERT.....	49
3.30 PROJEKT.DELETE.....	50
3.31 SERIENNUMMER.GET.....	51
3.32 SERIENNUMMER.PUT.....	53
3.33 SERIENNUMMER.INSERT.....	54
3.34 SERIENNUMMER.DELETE.....	55
3.35 CHARGE.GET.....	56
3.36 CHARGE.PUT.....	58
3.37 CHARGE.INSERT.....	59
3.38 CHARGE.DELETE.....	60
3.39 ADRESSARTIKEL.GET.....	61
3.40 VERSANDART.GET.....	118
3.41 VERSANDART.PUT.....	119
3.42 VERSANDART.INSERT.....	120
3.43 VERSANDART.DELETE.....	121
3.44 IMPORT.EXEC.....	140
3.45 MEMTAB.EXEC.....	141
3.46 DUBLETTENPRUEFUNG.EXEC.....	145
4 Anhang.....	147
4.1 Beispiel für Inline-Positionsdaten bei BELEG.INSERT.....	147
5 Übersicht der Änderungen.....	149

Dieses Dokument beschreibt die Servicefunktionen, die innerhalb der WEBWARE Webservices zur Verfügung stehen.

1 Glossar

1.1 Allgemeine Begriffe

WWSVC	WEBWARE Services (WEB Services der WEBWARE)
WWKIS	0
WWSEAS	0
HTTP	0
ServicePunkt	0
Hersteller	Erstellt Anwendungen die über WW-SVC Schnittstelle ausgeführt werden. Der Hersteller wird mit einer eindeutigen 32-Byte-Hash-ID sowie Name definiert.
Anwendung	<p>Eine Anwendung besteht aus einer definierten Funktionsgruppe und Rahmenparametern.</p> <p>Eine Anwendung wird mit einer eindeutigen 32-Byte-Hash-ID definiert.</p>
Secured App	Eine Secured App wird durch Freigabe/Aktivierung einer Hersteller/Anwendung auf einem WW-System erstellt. Diese enthält Rahmenparameter/Zugriffsbeschränkungen und wird mit einer Secure-APP-ID (Zahl) zusätzlich zu Hersteller/Anwendung beschrieben.
ServicePass	Möchte ein Client auf eine Secured App zugreifen, so muss er zuerst einen ServicePass beantragen. Der ServicePass besteht aus einem öffentlichen (PASS-ID) und geheimen (APP-SECRET) Hash-Wert.
PASS-ID	Dies ist ein öffentlicher Hash-Wert der den Zugriff auf eine Secured App ermöglicht.
APP-SECRET	Dies ist ein geheimer Hash-Wert der bei Registrierung an den

	Client übergeben wird.
	Dieser Hash-Wert wird danach nicht mehr innerhalb der Kommunikation verwendet.
Cookies	Ist eine Erweiterung des HTTP-Headers der notwendige/optionale Informationen enthält
Funktionsgruppe	Ist die Zusammenfassung von Einzelfunktionen und definiert auf welche Funktionen und mit welcher Berechtigung zugegriffen werden darf.
Funktion	Eine Funktion ist mit einem Namen beschrieben und kann über ein Aufruf über den ServicePunkt ausgeführt werden. (../WWSVC/[PASS-ID]/Funktions-Name..)
Parameter	Parameter können direkt beim Aufruf in der URL (URL-encoded) übergeben werden. ../WWSVC/[PASS-ID]/MeineFunktion/10001/SORT=KNDNR Parameter können auch direkt im Body der WWSVC-Anfrage übergeben werden.
JSON	JavaScript Object Notation: Datenformat im Browser-Bereich
XML	eXtended Markus Language: Datenformat im Browser-Bereich
Benutzerliste	Eine Liste von Benutzer-Namen und Passwörtern mit der der Zugriff bei Registrierung und oder Funktionszugriff geschützt werden kann.
iWWSVC-JS	Entwickler-Interface Test-Implementierung für WWSVC mit JavaScript Test: https://meine-webware.de/svc-js/svcjs.html

1.2 Zugriffsarten für WEBSERVICES

SYNCHRON	Ergebnis der Service-Funktion wird mit der gleichen Verbindung zurückgegeben
ASYNCHRON	0

ASYNCHRON_NO_RESULT 0

2 Allgemeines zu Servicefunktionen

2.1 Namenskonventionen für Objekte, Funktionen und Parameter

Das Namensschema ist stets RESOURCE.FUNKTION. Eine RESOURCE ist dabei eine Datenbanktabelle, ein Workflowskript, eine Auswertung oder ein spezielles Funktionsobjekt. FUNKTION ist der Name der Funktion, die auf der Ressource ausgeführt wird. Alle Ressourcen besitzen die parameterlose Funktion OPTIONS, die eine Liste der für die Ressource implementierten Funktionen liefert.

Verwendete Zeichen für Namen sind A-Z, a-z, 0-9 sowie der Unterstrich. Das erste Zeichen eines Namens darf A-Z, a-z und Unterstrich sein. Objekt- und Funktionsname werden durch einen Punkt getrennt (z.B. ARTIKEL.GET, hier ist ARTIKEL der Objektname, GET der Funktionsname).

Die maximale Länge von Namen beträgt jeweils 60 Zeichen.

Alle Namen werden intern in Großschreibung umgewandelt, das bedeutet, dass „ARTIKEL“, „Artikel“ und „artikel“ alle dasselbe Objekt beschreiben. **BETA-HINWEIS:** dies ist noch nicht getestet.

2.2 Hinweise zu freien Selektionsausdrücken

Der Parser für Selektionsausdrücke versteht geklammerte Boole'sche Ausdrücke mit den logischen Verknüpfungen & (UND) und # (ODER). Als Vergleichsoperatoren werden >, >=, =, <= und < erkannt, außerdem werden für (Teil-)Ausdrücke mit numerischen Operanden die arithmetischen Operatoren +, -, * und / sowie % (Modulo = Rest einer Division) ausgewertet, damit können auch einfache Berechnungen innerhalb eines Ausdrucks durchgeführt werden. Alphanumerische Literale müssen in einfache Hochkommas gesetzt werden.

Beispiele:

```
BEL_0_1 = 0 & BEL_2_1 = 'R'
ART_759_10 >= ART_769_10 + 1
ART_1_25 >= 'ArtikelChargennummer' # ART_36_5 = 'WGR01'
```

2.3 Hinweise zur Debug-Unterstützung

Bei allen Servicefunktionen kann ein Parameter MIT_DEBUGINFO angegeben werden. Mit Hilfe dieses Parameters werden im Ergebnissatz und/oder einer Protokolldatei Informationen ausgegeben, die Sie beim Debuggen Ihrer Anwendung unterstützen können.

Mit dem zweiten Parameter DEBUG_LEVEL kann gesteuert werden, wie detailliert die Debug-Ausgabe ausfällt. Hier können Werte von 0 (kein Protokoll) bis 4 (maximale Details) angegeben werden.

Parameter	Beschreibung
MIT_DEBUGINFO	0: keine Debug-Ausgabe 1: Debug-Ausgabe in der Rückgabe im Bereich DEBUGINFO 2: Debug-Ausgabe in eine Datei in LOGFILES\WWSVC 3: Debug-Ausgabe in Rückgabe und in Datei
DEBUG_LEVEL	0: keine Ausgabe 1: nur Fehler 2: nur Warnungen 3: Funktions-Trace 4: alle Ausgaben aktiv

Beispiel (Debug-Ausgabe in Rückgabe):

```
"COMRESULT": {
  "STATUS": 200,
  "CODE": "200 OK",
  "INFO": "Kein Fehler",
  "INFO2": "",
  "INFO3": ""
},
"DEBUGINFO": {
  "ROWFILTER": "(ART_1_25 >= 'Artikel_Sonderkonditionen') &
               (ART_1_25 <= 'Artikel_Sonderkonditionen')",
  "LOG": [
    "Ich bin ein Protokolleintrag!",
    "Ich auch."
  ]
},
"ARTIKELLISTE": {
  "ARTIKEL": [ ....
```

Ist Protokollierung in eine Datei aktiviert, so wird pro Abfrage eine Protokolldatei im Ordner LOGFILES\WWSVC im WEBWARE-Stammverzeichnis abgelegt. Der Dateiname setzt sich aus dem Präfix dbg_wwsvc_, der Prozess-ID des WWSVC-Prozesses und der laufenden Abfragenummer zusammen. Beispiel: dbg_wwsvc_13532_2.log. Bitte beachten Sie, dass im laufenden Betrieb so eine große Anzahl von Protokolldateien erzeugt werden können.

Es ist möglich, die Debug-Ausgabe per Steuerdatei permanent zu aktivieren. Dazu muss im Stammverzeichnis der WEBWARE eine Datei mit dem Namen `debug_wwsvc.ini` angelegt werden, die wie im nachfolgenden Beispiel aufgebaut ist. Die gültigen Werte für die beiden Parameter entsprechen der Übergabe per Webservice-Aufruf.

```
[Logging]
MitDebugInfo=2
DebugLevel=4
```

Ist eine Steuerdatei vorhanden und es werden zusätzlich die Debug-Parameter per Webservice-Aufruf übergeben, so gelten die Werte aus dem Aufruf.

Anmerkung: Der Umfang der Debug-Information wird bei Bedarf sukzessive erweitert.

2.4 Standardparameter

Es gibt eine Reihe von Standardparametern, die insbesondere bei den GET-Funktionen verwendet werden können. Welche Standardparameter von einer Funktion konkret unterstützt werden, ist bei der jeweiligen Funktion vermerkt.

Parameter	Funktion	Beschreibung
FELDER	GET	<p>übergibt eine Komma-getrennte Liste der abzurufenden Datenfelder. Beispiel:</p> <pre>FELDER=ART_1_25,ART_51_60</pre>
NUR_ANZAHL	GET	1: nur die Anzahl der ermittelten Datensätze wird zurückgeliefert.
NUR_GROESSE	GET	1: nur die Größe (in Bytes) des Antwortpakets wird zurückgeliefert.
SUCHE_VOLLTEXT	GET	<p>Übergibt einen Volltextsuchbegriff. Es wird dann im ersten Schritt die Volltextsuche nach dem Begriff ausgeführt, auf deren Ergebnis dann die weiteren Suchparameter angewendet.</p>
FREISELEKT	GET	<p>Übergibt einen freien Selektionsausdruck. Beispiel:</p> <pre>ART_36_5='WGR1' # ART_36_5='WGR2'</pre>
FREISELEKT_KEY	GET	<p>Übergibt die Nummer des Index, über den bei der gedreht wird.</p> <p>Bei Angabe von „AUTO“ werden ein passender Index und der Wertebereich automatisch anhand des Suchausdrucks ermittelt.</p>
FREISELEKT_VON_INDEX	GET	Übergibt den Startindex für den angegebenen Schlüssel. Der Schlüssel muss korrekt formatiert übergeben werden.
FREISELEKT_BIS_INDEX	GET	Übergibt den Endindex für den angegebenen Schlüssel. Der Schlüssel muss korrekt formatiert übergeben werden.
SELEKT_POOL_ID	GET	ID eines Selektionspools. Es wird dann im ersten Schritt über den Selektionspool iteriert, danach die weiteren Selektionsparameter

		angewandt.
SELEKT_POOL_BDNR	GET	Nummer des Bedieners, auf den sich SELEKT_POOL_ID bezieht. Zwingend erforderlich, wenn SELEKT_POOL_ID angegeben wird.
SELEKT_POOL_MATCHCODE	GET	Matchcode eines Selektionspool. Alternative Angabe zu SELEKT_POOL_ID.
FREISORT	GET	<p>Übergibt eine Komma-getrennte Liste der gewünschten Sortierfelder und optional der Sortierrichtung in der Form <Feldname>[ASC DESC][,...].</p> <p>Es können derzeit maximal fünf Felder angegeben werden.</p> <p>Beispiel:</p> <pre>FREISORT=ART_178_9 DESC,ART_1_25 ASC</pre>
MIT_LANGTEXT	GET	<p>Übergibt eine Liste von Langtextfeldern, die zurückgeliefert werden sollen.</p> <p>Für gültige Werte siehe die jeweilige GET-Funktion.</p>
OHNE_LEERFELDER	GET	1: Felder ohne Wert werden in der Rückgabe ausgelassen.
INTERNATIONALES_ZAHLEN FORMAT	GET	1: Liefert Gleitkommazahl mit Dezimalpunkt statt Komma.
DATUM_FORMAT	GET	<p>Übergibt eine Formatierungsanweisung für Datum-Felder.</p> <p>Folgende Tags werden erkannt:</p> <p>%t = Tag, %T = Tag 2-stellig %m = Monat, %M = Monat 2-stellig %j = Jahr 2-stellig, %J = Jahr 4-stellig</p> <p>Beispiel: %T.%M.%j für „07.03.19“ %J-%M-%T für „2019-03-07“</p>
OHNE_STAMMKALK	PUT INSERT	1: Es wird vor dem Schreiben keine Stammdatenkalkulation ausgeführt.
Name eines Langtextfeldes	PUT	Übergibt Langtext für das angegebene Feld.

NUR_TESTEN	INSERT INSERT 1: Nur testen, es wird kein Datensatz angelegt.
SELEKT_EVENTLOG	GET Eine Komma-getrennte Liste von Parametern: Art, VonDatum, VonUhrzeit, BisDatum, BisUhrzeit, <reserviert>, Feldselektion <i>Art:</i> NEU = liefere im Zeitraum neu angelegte Datensätze GEAENDERT = liefere die im Zeitraum geänderten Datensätze GELOESCHT = liefere die Primärschlüssel der im Zeitraum gelöschten Datensätze GELOESCHT_DURCH_WANDLUNG = liefere die Primärschlüssel der durch die Belegwandlung gelöschten Datensätze <i>VonDatum/BisDatum:</i> Beginn/Ende des Selektionszeitraums im Format DD.MM.JJJJ <i>VonUhrzeit/BisUhrzeit:</i> Beginn/Ende des Selektionszeitraums im Format HH:MM:SS <i>reserviert:</i> reserviert für SoftENGINE <i>Feldselektion (nur bei Art=GEAENDERT):</i> im Standard werden die im Ereignisprotokoll als geändert vermerkten Felder geliefert. Wird hier „ALLE_FELDER_LIEFERN“ übergeben, wird stattdessen der komplette Satz geliefert; falls vorhanden, wie die Feldliste aus dem Parameter FELDER verwendet.

2.5 Rückgabewerte von Servicefunktionen

Die Webservice-Funktionen liefern, mit wenigen Ausnahmen, eine JSON-Struktur zurück, die wenigstens ein „COMRESULT“-Objekt enthält. Enthält das STATUS-Feld den Wert 200, ist war der Funktionsaufruf erfolgreich, andernfalls ist der Wert des Feldes ein HTTP-Fehlercode und in den Feldern CODE sowie INFO, INFO2 und INFO3 sind nähere Angaben zur Fehlerursache enthalten.

```
"json":
{
  "COMRESULT":
  {
```

```

    "STATUS": 200,
    "CODE": "200 OK",
    "INFO": "Kein Fehler",
    "INFO2": "",
    "INFO3": ""
  }
}

```

Diese Struktur kann weitere Objekte enthalten, je nach Servicefunktion.

2.6 Standard-Rückgabe von GET-Funktionen

Bei GET-Funktionen wird in den meisten Fällen noch ein LISTE-Objekt enthalten sein. Der Name des LISTE-Objekts wird aus dem Namen der Ressource gebildet und lautet stets „<Ressourcenname>LISTE“, also z.B. „ARTIKELLISTE“ bei der ARTIKEL-Ressource. Das LISTE-Objekt enthält ein ganzzahliges Feld ANZAHL, welches die Anzahl der gelieferten Datensätze enthält und ein Array, das die zurückgelieferten Datensätze enthält. Das Array heißt dabei immer genauso wie die Ressource.

Im nachfolgenden Beispiel ist ARTIKELLISTE das Rückgabeobjekt. Es enthält ein Array (ARTIKEL) von Objekten und das Feld ANZAHL. Die Objekte des Arrays wiederum bestehen aus dem Feld SNR, das die Satznummer des Satzes enthält, und die über den FELDER-Parameter angegebenen Feldnamen und Feldwerte.

Das LISTE-Objekt enthält ferner stets das Feld ANZAHL, das die Anzahl der gelieferten Array-Elemente enthält.

```

"json":
{
  "COMRESULT":
  {
    "STATUS": 200,
    "CODE": "200 OK",
    "INFO": "Kein Fehler",
    "INFO2": "",
    "INFO3": ""
  },
  "ARTIKELLISTE":
  {
    "ARTIKEL":
    [
      {
        "SNR": 123,
        "ART_1_25": "Artikel1",
        "ART_51_60": "Artikelbezeichnung1",
        ...
      },
      ...
    ],
    "ANZAHL": <Anzahl gelieferter Datensätze>
  }
}

```

Die Array-Elemente können weitere Unterobjekte enthalten. Diese sind bei der jeweiligen GET-Funktion beschrieben.

Werden die Parameter NUR_ANZAHL oder NUR_GROESSE verwendet, sieht das Ergebnisobjekt wie folgt aus.

Rückgabe (bei Angabe von NUR_ANZAHL = 1):

```
{  
  "ANZAHL": <Anzahl der Datensätze>  
}
```

Rückgabe (bei Angabe von NUR_GROESSE = 1):

```
{  
  "BYTES": <Größe in Bytes>  
}
```


3 Referenz Servicefunktionen

In diesem Abschnitt finden Sie eine Referenz der verfügbaren Servicefunktionen. Bitte beachten: Während der BETA-Phase sind jederzeit Änderungen bei Ein- und Ausgabeformaten sowie der Namen der Funktionen und Ressourcen möglich.

3.1 ARTIKEL.GET

Holt eine Liste von Artikeldatensätzen.

Parameter (optional):

Parameter	Beschreibung
ARTNR	
VON_ARTNR	
BIS_ARTNR	Schränkt die Liste auf den angegebenen Artikelnummernbereich ein.
WGR	
VON_WGR	
BIS_WGR	Schränkt die Liste auf den angegebenen Warengruppenbereich ein.
KATALOG	
VON_KATALOG	
BIS_KATALOG	Schränkt die Liste auf den angegebenen Katalognummernbereich ein.
MIT_KATEGORIEN	1: Liefert innerhalb des Artikelobjekts eine Liste der den Artikel enthaltenden Marketplace-Kategorien als Unterobjekt (Format siehe unten).
KATEGORIE_FELDER	Übergibt eine Komma-getrennte Liste der gewünschten Datenfelder der Kategorientabelle (KAT_0_10, etc.).
MIT_LAGERBESTAND	1: Ruft Lagerbestände in die Fehler ART_759_10 und ART_769_10 ab.
LAGER	
VON_LAGER	
BIS_LAGER	Schränkt Lager für Lagerbestandsermittlung auf den angegebenen Bereich ein.

MIT_LAGERPLAETZEN	1: Ruft Lagerbestände auf Lagerplätzen als Unterobjekt ab.
LAGERPLATZ_FELDER	Übergibt Komma-getrennte Liste der gewünschten Felder der Lagerplatztabelle (ARTLG_27_8, ARTLG_35_12,...).
MIT_ATTRIBUTEN	1: Ruft alle zum Artikel hinterlegten Attribute ab.
ATTRIBUT_FELDER	Übergibt Komma-getrennte Liste der gewünschten Felder der Attributtabelle (ATR_61_10, ATR_88_256,...).
MIT_ATTRIBUT_LANGTEXT	Übergibt Parameter für Langtext-Attribute. Dies ist eine Komma-getrennte Liste folgender möglicher Werte: LT00, LT01, LT02, LT03, LT04.
MIT_ATTRIBUTDEFINITION	1: Ruft zu jedem Attribut auch dessen Attributdefinition ab.
ATTRIBUTDEFINITION_FELDER	Übergibt Komma-getrennte Liste der gewünschten Felder Attributdefinitionstabelle (ATD_10_60,...).
MIT_VARIANTEN	1: Ist ein Artikel ein Variantenhauptartikel, werden die Artikelnummern seiner Varianten als Liste mit ausgegeben (siehe Beispiel).
MIT_VARIANTEN_WERTE	1: Für alle Variantenartikel werden die Ausprägungen mit ausgegeben (siehe Beispiel).

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	
NUR_ANZAHL	
NUR_GROESSE	
SUCHE_VOLLTEXT	
FREISELEKT	
FREISELEKT_KEY	
FREISELEKT_VON_INDEX	
FREISELEKT_BIS_INDEX	

Parameter	Erlaubte Werte/Anmerkung
FREISORT	
MIT_LANGTEXT	NOTIZTEXT, LANGTEXT, WARNTTEXT, KURZTEXT, LT01, LT02, LT03, LT04, LT05
OHNE_LEERFELDER	

Rückgabe:

```
ARTIKELLISTE:
{
  ARTIKEL:
  [
    {
      SNR: <Satznummer>,
      Felder gemäß Feldliste oder alle Felder
    }, ...
  ],
  ANZAHL: <Anzahl der Datensätze>
}
```

Rückgabe (Unterobjekt Kategorien, bei Angabe von MIT KATEGORIEN = 1):

```
KATEGORIENLISTE:
{
  KATEGORIE:
  [
    {
      Felder gemäß Feldliste oder alle Felder
    }, ...
  ],
  ANZAHL: <Anzahl der Datensätze>
}
```

Rückgabe (Unterobjekt Varianten, bei Angabe von MIT VARIANTEN = 1):

```
VARIANTENLISTE:
{
  VARIANTE:
  [
    {
      ARTNR: <Artikelnummer>
      WERTE
      {

```

```

        <Attributname>: <Attributwert>,
        ...
    }
},
...
]
}

```

Rückgabe (Unterobjekt Lagerplätze, bei Angabe von MIT_LAGERPLAETZEN=1):

```

LAGERPLATZLISTE:
{
  LAGERPLATZ:
  [
    {
      Felder gemäß Feldliste oder alle Felder
    }, ...
  ]
}

```

3.2 ARTIKEL.PUT

Aktualisiert einen vorhandenen Artikel

Parameter (obligatorisch):

Parameter	Beschreibung
ARTNR	Artikelnummer
ART_...	zu setzende Felder

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe ARTIKEL.GET
OHNE_STAMMKALK	

Rückgabe:

Keine

3.3 ARTIKEL.INSERT

Legt einen neuen Artikel an.

Parameter (obligatorisch):

Parameter	Beschreibung
ART_...	zu setzende Felder

Parameter (optional):

Parameter	Beschreibung
ARTNR	Vorgabe für die Artikelnummer, bei automatischer Vergabe Vorlage für die Artikelnummer
ERFASSUNGSGRUPPE	zu verwendende Artikelerfassungsgruppe
ARTNR_AUTOMATIK	1: verwende automatische Artikelnummervergabe, ARTNR ist dann die Vorlage.

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe ARTIKEL.GET
OHNE_STAMMKALK	
NUR_TESTEN	

Rückgabe:

```
{  
  ARTNR: <die neu vergebene Artikelnummer>  
}
```


3.4 ARTIKEL.DELETE

Löscht einen Artikeldatensatz.

Parameter (obligatorisch):

Parameter	Beschreibung
ARTNR	die Artikelnummer des zu löschenden Artikels

Rückgabe:

Keine

Hinweis:

Es wird eine Überprüfung durchgeführt, ob der Artikel gelöscht werden kann. Artikel mit Umsatz können nicht gelöscht werden.

3.5 ADRESSE.GET

Holt eine Liste von Adressdatensätzen.

Parameter (optional):

Parameter	Beschreibung
ADNR	
VON_ADRNR	
BIS_ADRNR	Schränkt die Liste auf den angegebenen Adressnummernbereich ein.
MIT_ANSPRECHPARTNERN	1: Ruft eine Liste der der Adresse zugeordneten Ansprechpartner als Subobjekt ab.
VON_ANPNR	
BIS_ANPNR	Schränkt die Liste der Ansprechpartner auf den angegebenen Ansprechpartnernummernbereich ein.
ANSPRECHPARTNER_FELDER	Übergibt Komma-getrennte Liste der gewünschten Felder der Ansprechpartnerliste (ANP_).
MIT_LIEFERADRESSEN	1: Ruft eine Liste der der Adresse zugeordneten Lieferadressen als Subobjekt ab.
VON_LFANR	
BIS_LFANR	Schränkt die Liste der Lieferadressen auf den angegebenen Lieferadressnummernbereich ein.
LIEFERADDRESS_FELDER	Übergibt Komma-getrennte Liste der gewünschten Felder der Lieferadressliste (LFA_).

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	

NUR_ANZAHL	
NUR_GROESSE	
SUCHE_VOLLTEXT	
FREISELEKT	
FREISELEKT_KEY	
FREISELEKT_VON_INDEX	
FREISELEKT_BIS_INDEX	
FREISORT	
MIT_LANGTEXT	NOTIZTEXT, WARNTTEXT, NT72, NT73, NT74, NT75, NT76
OHNE_LEERFELDER	

Rückgabe:

```

ADRESSLISTE
{
  ADRESSE
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}

```

Rückgabe (Unterobjekt Ansprechpartner):

```

ANSPRECHPARTNERLISTE
{
  ANSPRECHPARTNER
  [
    {

```

```

        SNR: <Satznummer>
        Felder gemäß Feldliste oder alle Felder
    },
    ...
]
ANZAHL: <Anzahl gelieferter Datensätze>
}

```

Rückgabe (Unterobjekt Lieferadressen):

```

LIEFERADRESSLISTE
{
    LIEFERADRESSE
    [
        {
            SNR: <Satznummer>
            Felder gemäß Feldliste oder alle Felder
        },
        ...
    ]
    ANZAHL: <Anzahl gelieferter Datensätze>
}

```

3.6 ADRESSE.PUT

Aktualisiert einen bestehenden Adressdatensatz

Parameter (obligatorisch):

Parameter	Beschreibung
ADRNR	Adressnummer
ADR_...	zu setzende Felder

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe ADRESSE.GET.
OHNE_STAMMKALK	

Rückgabe:

Keine

3.7 ADRESSE.INSERT

Erstellt einen neuen Adressdatensatz.

Parameter (obligatorisch):

Parameter	Beschreibung
TYP	Mögliche Werte: KUNDE, LIEFERANT, ERSTKONTAKT
ADR_...	Zu setzende Felder.

Parameter (optional):

Parameter	Beschreibung
ADRNr	Vorgabe für Adressnummer, leer lassen für automatische Vergabe.
ADRART	Vorgabe Adressart
ADRKREIS	Vorgabe Adresskreis
LAND	Vorgabe Ländereinstellungen

Verwendbare Standardparameter:

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe die ADRESSE.GET
OHNE_STAMMKALK	

Rückgabe:

```
{
```

```
ADRNR: <Adressnummer des neu erstellen Datensatzes>  
}
```

3.8 ADRESSE.DELETE

Löscht einen Adressdatensatz.

Parameter (obligatorisch):

Parameter	Beschreibung
ADRNR	Adressnummer

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Adressdatensatz gelöscht werden kann.

3.9 BELEG.GET

Holt eine Liste von Belegkopfdatensätzen

Parameter (optional):

Parameter	Beschreibung
VON_ADRNR	
BIS_ADRNR	Schränkt die Liste auf den angegebenen Adressnummernbereich ein.
BELNDX	
VON_BELNDX	
BIS_BELNDX	Schränkt die Liste auf den angegebenen Belegindexbereich ein.
JAHR	
VON_JAHR	
BIS_JAHR	Schränkt die Liste auf das angegebene Geschäftsjahr ein (0 – 9).
BELART	Schränkt die Liste auf die angegebene Belegart ein.
BELNR	
VON_BELNR	
BIS_BELNR	Schränkt die Liste auf den angegebenen Belegnummernbereich ein.
MIT_POSITIONEN	1: liefert die Positionen jedes Belegs als Unterobjekt.
POSITION_FELDER	Übergibt eine Komma-getrennte Liste der gewünschten Datenfelder der Positionstabelle.

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	

NUR_ANZAHL

NUR_GROESSE

SUCHE_VOLLTEXT

FREISELEKT

FREISELEKT_KEY

FREISELEKT_VON_INDEX

FREISELEKT_BIS_INDEX

FREISORT

SELEKT_POOL_ID

SELEKT_POOL_BDNR

SELEKT_POOL_MATCHCODE

MIT_LANGTEXT BELTXT, BELNTXT, BELVTXT, BELTXT0 bis BELTXT9

OHNE_LEERFELDER

Rückgabe:

```
BELEGLISTE
{
  BELEG
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

Rückgabe (Unterobjekt Positionen, bei Angabe von MIT_POSITIONEN = 1):

```
POSITIONSLISTE:
{
  POSITION:
  [
    {
```

```
        Felder gemäß Feldliste oder alle Felder
    }, ...
],
ANZAHL: <Anzahl der Datensätze>
}
```

3.10 BELEG.PUT

Aktualisiert einen vorhandenen Beleg

Parameter (obligatorisch):

Parameter	Beschreibung
BELNDX	der Index des Belegs
BEL_...	Zu setzende Felder.

Parameter (optional):

Parameter	Beschreibung
BERECHNE_SUMMEN	1: Belegsummen werden neu berechnet
KEINE_VERBUCHUNG	1: es wird keine Verbuchung durchgeführt
KALKNDX	Abweichender Index der verwendeten Stammdatenkalkulation.
POS_DATENFEHLER_IGNORIEREN	1: Positionen werden auch dann aktualisiert, wenn die interne Prüfung einen fehlerhaften Satz erkennt.

Verwendbare Standardparameter:

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe die BELEG.GET

Rückgabe:

Keine

3.11 BELEG.INSERT

Erstellt einen Warenwirtschaftsbeleg

Parameter (obligatorisch):

Parameter	Beschreibung
BEL_2_1	Belegart
BEL_11_8	Adressnummer

Parameter (optional):

Parameter	Beschreibung
BEL_...	weitere Belegkopffelder
POS_DATENFEHLER_IGNORIEREN	1: Positionen werden auch dann angelegt, wenn die interne Prüfung einen fehlerhaften Satz erkennt.

Verwendbare Standardparameter:

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe die BELEG.GET

Übergabe von Positionsdaten:

Belegpositionsdaten können als JSON-Substruktur wie folgt übergeben werden:

```
POSDATEN:
[
  {
    PARAMETER:
    [
      {
        POS_18_25      Artikelnummer
        POS_164_8      Menge
        POS_...        weitere Positionsfelder
      },
      ...
    ]
  }
]
```

Ein detailliertes Beispiel findet sich im Anhang.

Rückgabe:

```
{
  BELNDX: <Index des neu angelegten Belegs>
  POS_ANGELEGT: <Anzahl angelegter Positionen>
}
```

3.12 BELEG.DELETE

Löscht einen Belegkopfsatz inklusive seiner Positionssätze.

Parameter (obligatorisch):

Parameter	Beschreibung
BELNDX	Primärschlüssel des zu löschenden Belegs

Parameter (optional):

Parameter	Beschreibung
KEINE_ARCHIVIERUNG	1: es wird keine Archivierung durchgeführt

Rückgabe:

Keine

3.13 BELPOS.GET

Holt die Positionen zu einem Beleg

Parameter (obligatorisch):

Parameter	Beschreibung
BELNDX	vollständiger Belegindex

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	
NUR_ANZAHL	
NUR_GROESSE	
MIT_LANGTEXT	NOTIZTEXT, NOTIZTEXT2, NOTIZTEXT3, NOTIZTEXT4
OHNE_LEERFELDER	
FREISORT	

Rückgabe:

```
POSITIONSLISTE
{
  POSITION
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
}
```



```
ANZAHL: <Anzahl gelieferter Datensätze>  
}
```

3.14 BELPOS.PUT

Aktualisiert einen vorhanden Belegpositionssatz

Parameter (obligatorisch):

Parameter	Beschreibung
SNR	Die Satznummer des zu aktualisierenden Satzes.

Parameter (optional):

Parameter	Beschreibung
MIT_VERBUCHUNG	1: Der Satz wird für die Statistik verbucht
MIT_BERECHNUNG	1: Die Formel wird ausgeführt.
POS_DATENFEHLER_IGNORIEREN	1: Die Position wird auch dann aktualisiert, wenn die interne Prüfung einen fehlerhaften Satz erkennt.

Verwendbare Standardparameter:

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe die BELPOS.GET

Rückgabe:

Keine

3.15 BELPOS.INSERT

Fügt eine neue Belegposition ein.

Parameter (obligatorisch):

Parameter	Beschreibung
BELNDX	Primärschlüssel des Belegs
POS_18_25	Artikelnummer
POS_164_8	Menge

Parameter (optional):

Parameter	Beschreibung
LANGTEXT_AUFLOESEN	1: Langtext wird aufgelöst
EINFUEGE_SNR	Satznummer des Vorgängers
POS_DATENFEHLER_IGNORIEREN	1: Die Position wird auch dann angelegt, wenn die interne Prüfung einen fehlerhaften Satz erkennt.

Verwendbare Standardparameter:

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe die DBK.GET

Rückgabe:

```
{
  SNR: <Satznummer des eingefügten Satzes>
}
```

```
}
```

3.16 IDB<IDBID>.GET

Holt eine Liste von Datensätzen aus der IDB mit der ID <IDBID>.

Parameter (optional):

Parameter	Beschreibung
PK	
VON_PK	
BIS_PK	schränkt die Liste auf den angegebenen Primärschlüsselbereich ein.

Verwendbare Standardparameter (optional):

FELDER	
NUR_ANZAHL	
NUR_GROESSE	
SUCHE_VOLLTEXT	
FREISELEKT	
FREISELEKT_KEY	
FREISELEKT_VON_INDEX	
FREISELEKT_BIS_INDEX	
FREISORT	
MIT_LANGTEXT	NOTIZTEXT1, NOTIZTEXT2 Aliase: NT1, NT2

OHNE_LEERFELDER	
-----------------	--

Rückgabe:

```
IDB<IDBID>LISTE
{
  IDB<IDBID>
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

3.17 IDB<IDBID>.PUT

Aktualisiert einen vorhandenen IDB-Satz.

Parameter (obligatorisch):

Parameter	Beschreibung
PK	Primärschlüssel
IDB_...	Zu setzende Felder.

Parameter (optional):

Parameter	Beschreibung
Name eines Langtextfeldes	gültige Namen siehe IDB.GET.
OHNE_STAMMKALK	1: keine Stammdatenkalkulation durchführen

Rückgabe:

Keine

3.18 IDB<IDBID>.INSERT

Erstellt einen neuen IDB-Satz.

Parameter (optional):

Parameter	Beschreibung
PK	Vorgabe für Primärschlüssel. Kann leer bleiben, falls in den IDB-Einstellungen die automatische Nummerierung aktiviert ist.
IDB_...	Weitere zu setzende Felder.

Verwendbare Standardparameter:

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe die ADRESSE.GET
OHNE_STAMMKALK	
NUR_TESTEN	

Rückgabe:

```
{  
  PK: <Primärschlüssel des neu erstellen Datensatzes>  
}
```


3.19 IDB<IDBID>.DELETE

Löscht einen IDB-Satz.

Parameter (obligatorisch):

Parameter	Beschreibung
PK	der Primärschlüssel des zu löschenden Datensatzes

Rückgabe:

Keine

3.20 DBK<DBKNR>.GET

Holt eine Liste von Datensätzen aus der DBK mit der Nummer <DBKNR>.

Parameter (optional):

Parameter	Beschreibung
PK	
VON_PK	
BIS_PK	schränkt die Liste auf den angegebenen Primärschlüsselbereich ein

Verwendbare Standardparameter (optional):

Parameter	Gültige Werte/Anmerkung
FELDER	
NUR_ANZAHL	
NUR_GROESSE	
FREISELEKT	
FREISELEKT_KEY	
FREISELEKT_VON_INDEX	
FREISELEKT_BIS_INDEX	
FREISORT	
MIT_LANGTEXT	NOTIZTEXT
	Alias: NT
OHNE_LEERFELDER	

Rückgabe:

```
DBK<DBKNR>LISTE
{
  DBK<DBKNR>
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

3.21 DBK<DBKNR>.PUT

Aktualisiert einen vorhandenen DBK-Satz.

Parameter (obligatorisch):

Parameter	Beschreibung
PK	Primärschlüssel
DBK_...	Zu setzende Felder.

Verwendbare Standardparameter:

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe die DBK.GET
OHNE_STAMMKALK	

Rückgabe:

Rückgabe:

Keine

3.22 DBK<DBKNR>.INSERT

Erstellt einen neuen DBK-Satz.

Parameter (obligatorisch):

NUR DBK25!

Parameter	Beschreibung
DBK_70_10	Stichtag
DBK_80_10	Aufnahmetag
DBK_90_8	Lager

Parameter (optional):

Parameter	Beschreibung
PK	Vorgabe für Primärschlüssel. Kann leer bleiben, falls die DBK-Tabelle eine automatische Nummerierung unterstützt.

Verwendbare Standardparameter:

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe die DBK.GET
OHNE_STAMMKALK	

Rückgabe:

```
{  
  PK: <Primärschlüssel des neu erstellen Datensatzes>  
}
```

Anmerkung für die DBK25:

Wenn der Parameter PK übergeben wird, muss der Wert mindestens 31 Zeichen lang sein und die Teilwerte **Nummer+Stichtag+Aufnahmetag+Lager** in korrekter Formatierung enthalten. Da im Regelfall die nächste freie Nummer unbekannt ist, können auch nur die Parameter DBK_70_10, DBK_80_10 und DBK_90_8 übergeben werden, woraufhin das System die nächste freie Nummer ermittelt, den Datensatz anlegt und den PK des Satzes zurückgibt. Das Feld PK muss in dem Fall leer gelassen werden.

3.23 DBK<DBKNR>.DELETE

Löscht einen DBK-Satz

Parameter (obligatorisch):

Parameter	Beschreibung
PK	der Primärschlüssel des zu löschenden Datensatzes

Rückgabe:

Keine

3.24 DBP<DBPNR>.GET

Holt die Positionen zu einem DBK-Satz

Parameter (obligatorisch):

Parameter	Beschreibung
DBKNDX	DBK-Primärschlüssel

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	
OHNE_LEERFELDER	
FREISORT	

Rückgabe:

```
DBP<DBPNR>LISTE
{
  DBP<DBPNR>
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```


3.25 DBP<DBPNR>.PUT

Aktualisiert einen vorhanden DBK-Positionssatz

Parameter (obligatorisch):

Parameter	Beschreibung
SNR	Die Satznummer des zu aktualisierenden Satzes
DBP_...	Zu setzende Felder.

Parameter (optional):

Parameter	Beschreibung
MIT_VERBUCHUNG	1: Der Satz wird für die Statistik verbucht
MIT_BERECHNUNG	1: Die Formel wird ausgeführt.

Rückgabe:

Keine

3.26 DBP<DBPNR>.INSERT

Fügt einen neuen DBK-Positionssatz ein.

Parameter (obligatorisch):

Parameter	Beschreibung
DBKNDX	Primärschlüssel des DBK-Satzes
DBP_...	Zu setzende Felder

Parameter (optional):

Parameter	Beschreibung
MIT_VERBUCHUNG	1: Der Satz wird für die Statistik verbucht.
MIT_BERECHNUNG	1: Die Formel wird ausgeführt.
EINFUEGE_SNR	Satznummer des Vorgängersatzes

Rückgabe:

```
{
  SNR:  <Satznummer des eingefügten Satzes>
}
```

3.27 PROJEKT.GET

Holt eine Liste von Projektdatensätzen

Parameter (optional):

Parameter	Beschreibung
PRJNR	
VON_PRJNR	
BIS_PRJNR	Schränkt die Liste auf den angegebenen Projektnummernbereich ein.
ADRNR	
VON_ADRNR	
BIS_ADRNR	Schränkt die Liste auf den angegebenen Adressnummernbereich ein.

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	
NUR_ANZAHL	
NUR_GROESSE	
SUCHE_VOLLTEXT	
FREISELEKT	
FREISELEKT_KEY	
FREISELEKT_VON_INDEX	
FREISELEKT_BIS_INDEX	
FREISORT	

Parameter	Erlaubte Werte/Anmerkung
MIT_LANGTEXT	NOTIZTEXT, WARNTXT Aliase: NT, WT
OHNE_LEERFELDER	

Rückgabe:

```
PROJEKTLISTE
{
  PROJEKT
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

3.28 PROJEKT.PUT

Aktualisiert einen vorhandenen Projektsatz

Parameter (obligatorisch):

Parameter	Beschreibung
PRJNR	Projektnummer
PRJ_...	Zu setzende Felder

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe PROJEKT.GET
OHNE_STAMMKALK	

Rückgabe:

Keine

3.29 PROJEKT.INSERT

Erstellt einen neuen Projektsatz

Parameter (optional):

Parameter	Beschreibung
PRJNR	Vorgabe für Projektnummer, leer lassen für automatische Vergabe.

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe PROJEKT.GET
OHNE_STAMMKALK	
NUR_TESTEN	

Rückgabe:

```
{  
  PRJNR: <Projektnummer des neu erstellen Datensatzes>  
}
```

3.30 PROJEKT.DELETE

Löscht einen Projektdatensatz.

Parameter (obligatorisch):

Parameter	Beschreibung
PRJNR	Projektnummer

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Projektdatensatz gelöscht werden kann.

3.31 SERIENNUMMER.GET

Holt eine Liste von Seriennummern

Parameter (optional):

Parameter	Beschreibung
SERNR	
VON_SERNR	
BIS_SERNR	Schränkt die Liste auf den angegebenen Seriennummernbereich ein.
ARTNR	
VON_ARTNR	
BIS_ARTNR	Schränkt die Liste auf den angegebenen Artikelnummernbereich ein.

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	
NUR_ANZAHL	
NUR_GROESSE	
SUCHE_VOLLTEXT	
FREISELEKT	
FREISELEKT_KEY	
FREISELEKT_VON_INDEX	
FREISELEKT_BIS_INDEX	
FREISORT	

Parameter	Erlaubte Werte/Anmerkung
MIT_LANGTEXT	NOTIZTEXT, WARNTXT Aliase: NT, WT
OHNE_LEERFELDER	

Rückgabe:

```
SERIENNUMMERNLISTE
{
  SERIENNUMMER
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

3.32 SERIENNUMMER.PUT

Aktualisiert einen vorhandenen Seriennummernsatz.

Parameter (obligatorisch):

Parameter	Beschreibung
SERNR	Seriennummer
SER_...	Zu setzende Felder

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe SERIENNUMMER.GET
OHNE_STAMMKALK	

Rückgabe:

Keine

3.33 SERIENNUMMER.INSERT

Erstellt einen neuen Seriennummernsatz.

Parameter (obligatorisch):

Parameter	Beschreibung
SERNR	Seriennummer

Parameter (optional):

Parameter	Beschreibung
SER_...	zu setzende Felder

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe SERIENNUMMER.GET
OHNE_STAMMKALK	
NUR_TESTEN	

Rückgabe:

```
{
  SERNR: <Primärschlüssel des neu erstellen Datensatzes>
}
```

3.34 SERIENNUMMER.DELETE

Löscht einen Seriennummernsatz.

Parameter (obligatorisch):

Parameter	Beschreibung
SERNR	Seriennummer

Rückgabe:

Keine

3.35 CHARGE.GET

Holt eine Liste von Chargen

Parameter (optional):

Parameter	Beschreibung
CHANR	
VON_CHANR	
BIS_CHANR	Schränkt die Liste auf den angegebenen Chargennummernbereich ein.
ARTNR	
VON_ARTNR	
BIS_ARTNR	Schränkt die Liste auf den angegebenen Artikelnummernbereich ein.

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	
NUR_ANZAHL	
NUR_GROESSE	
SUCHE_VOLLTEXT	
FREISELEKT	
FREISELEKT_KEY	
FREISELEKT_VON_INDEX	
FREISELEKT_BIS_INDEX	
FREISORT	

Parameter	Erlaubte Werte/Anmerkung
MIT_LANGTEXT	NOTIZTEXT, WARNTXT Aliase: NT, WT
OHNE_LEERFELDER	

Rückgabe:

```
CHARGENLISTE
{
  CHARGE
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

3.36 CHARGE.PUT

Aktualisiert einen vorhandenen Chargensatz.

Parameter (obligatorisch):

Parameter	Beschreibung
CHANR	Chargennummer
CHA_...	Zu setzende Felder

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe CHARGE.GET
OHNE_STAMMKALK	

Rückgabe:

Keine

3.37 CHARGE.INSERT

Erstellt einen neuen Chargensatz.

Parameter (obligatorisch):

Parameter	Beschreibung
CHANR	Chargennummer

Parameter (optional):

Parameter	Beschreibung
CHA_...	zu setzende Felder

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
Name eines Langtextfeldes	gültige Namen siehe CHARGE.GET
OHNE_STAMMKALK	
NUR_TESTEN	

Rückgabe:

```
{  
  CHANR: <Primärschlüssel des neu erstellen Datensatzes>  
}
```


3.38 CHARGE.DELETE

Löscht einen Chargensatz.

Parameter (obligatorisch):

Parameter	Beschreibung
CHANR	Chargennummer

Rückgabe:

Keine

3.39 ADRESSARTIKEL.GET

Holt eine Liste von Artikeldatensätzen.

Parameter (optional):

- ADRNR
- VON_ADRNR
- BIS_ADRNR schränkt die Liste auf den angegebenen Adressnummernbereich ein
- ARTNR
- VON_ARTNR
- BIS_ARTNR schränkt die Liste auf den angegebenen Artikelnummernbereich ein

Verwendbare Standardparameter (optional):

- FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder
(z.B. FELDER=ADA_1_8,ADA_9_25)
- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen Datensätze
- NUR_GROESSE 1=Liefert als Antwortpaket nur die Größe des in Bytes des tatsächlichen Antwortpakets
- SUCHE_VOLLTEXT übergibt einen Volltext-Suchbegriff. Die Volltextsuche wird als erster Schritt durchgeführt, danach werden weitere Selektionsparameter angewandt.
- FREISELEKT übergibt einen freien Selektionsausdruck.
Bsp.: (ADA_34_25 = '123' & ADA_59_8 > 10)
- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n) oder "AUTO" für automatische Ermittlung aus dem Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX Startwert für Index
- FREISELEKT_BIS_INDEX Endwert für Index

- **FREISORT** übergibt eine Komma-getrennte Liste der gewünschten Sortierfelder
FREISORT = <Feldname>[ASC|DESC][,...]
 (z.B. **FREISORT**=ART_178_9 DESC,ART_1_25 ASC)
 Es können derzeit maximal fünf Felder angegeben werden.
- **MIT_LANGTEXT** übergibt Langtext-Parameter. Dies ist eine Komma-getrennte Liste folgender möglicher Werte: NOTIZTEXT und NOTIZTEXT2
- **OHNE_LEERFELDER** 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett ausgelassen.

Rückgabe (normal):

```
ADRESSARTIKELLISTE
{
    ADRESSARTIKEL
    [
        {
            SNR: <Satznummer>
            Felder gemäß Feldliste oder alle Felder
        },
        ...
    ]
    ANZAHL: <Anzahl gelieferter Datensätze>
}
```

Rückgabe (NUR_ANZAHL):

```
{
    ANZAHL: <Anzahl der Datensätze>
}
```

Rückgabe (NUR_GROESSE):

```
{
    BYTES: <Größe in Bytes>
}
```

3.40 ADRESSARTIKEL.PUT

Aktualisiert einen vorhandenen Adressartikel

Parameter (obligatorisch):

- ADRNR Adressnummer
- ARTNR Artikelnummer

Parameter (optional):

- ADA_... zu setzende Felder
- OHNE_STAMMKALK 1 = keine Stammdatenkalkulation durchführen
- Name eines Langtextfeldes gültige Namen siehe die GET-Funktion

Rückgabe:

Keine

3.41 ADRESSARTIKEL.INSERT

Legt einen neuen Adressartikel an.

Parameter (obligatorisch):

- ADRNR Adressnummer
- ARTNR Artikelnummer

Parameter (optional):

- ADA_... zu setzende Felder
- Name eines Langtextfeldes gültige Namen siehe die GET-Funktion

Rückgabe:

```
{
  ADANDX: <Primärschlüssel des neu angelegten Datensatzes>
}
```

3.42 ADRESSARTIKEL.DELETE

Löscht einen Adressartikelsatz.

Parameter (obligatorisch):

- ADRNR Adressnummer
- ARTNR Artikelnummer

Rückgabe:

Keine

Hinweis: es wird eine Überprüfung durchgeführt, ob der Datensatz gelöscht werden kann. Abhängig von Zugriffsrechten kann es sein, dass ein Datensatz nicht gelöscht werden kann.

3.43 LIEFERADRESSE.GET

Holt eine Liste von Lieferadressdatensätzen.

Parameter (optional):

- ADRNR
- VON_ADRNR
- BIS_ADRNR schränkt die Liste auf den angegebenen Adressnummernbereich ein
- LFANR
- VON_LFANR
- BIS_LFANR schränkt die Liste auf den angegebenen Lieferadressnummernbereich ein

Verwendbare Standardparameter (optional):

- FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder (z.B. FELDER=LFA_2_8,LFA_10_8)
- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen Datensätze
- NUR_GROESSE 1=Liefert als Antwortpaket nur die Größe des in Bytes des tatsächlichen Antwortpakets
- FREISELEKT übergibt einen freien Selektionsausdruck.
Bsp.: (LFA_110_10 >= 66000 # LFA_110_10 <= 66999)
- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n) oder "AUTO" für automatische Ermittlung aus dem Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX
Startwert für Index
- FREISELEKT_BIS_INDEX
Endwert für Index
- FREISORT übergibt eine Komma-getrennte Liste der gewünschten Sortierfelder
FREISORT = <Feldname>[ASC|DESC][,...]
(z.B. FREISORT=ART_178_9 DESC,ART_1_25 ASC)
Es können derzeit maximal fünf Felder angegeben werden.

Rückgabe:

```
LIEFERADRESSLISTE
{
  LIEFERADRESSE
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

Rückgabe (NUR_ANZAHL):

```
{
  ANZAHL: <Anzahl der Datensätze>
}
```

Rückgabe (NUR_GROESSE):

```
{
  BYTES: <Größe in Bytes>
}
```


3.44 LIEFERADRESSE.PUT

Aktualisiert einen bestehenden Lieferadressensatz

Parameter (obligatorisch):

- LFANR Lieferadressnummer

Parameter (optional):

- LFA_... zu setzende Felder
- OHNE_STAMMKALK 1=keine Stammdatenkalkulation durchführen
- Name eines Langtextfeldes gültige Namen siehe die GET-Funktion

Rückgabe:

Keine

3.45 LIEFERADRESSE.INSERT

Erstellt einen neuen Lieferadressdatensatz.

Parameter (optional):

- LFA_... zu setzende Felder
- LFANR Vorgabe für Lieferadressnummer
- OHNE_STAMMKALK 1 = Keine Stammdatenkalkulation durchführen
- Name eines Langtextfeldes gültige Namen siehe die GET-Funktion

Rückgabe:

```
{  
    LFANR: <Lieferadressnummer des neu erstellen Datensatzes>  
}
```

Hinweis: wird der Parameter LFANR nicht mitgeliefert, wird eine neue Lieferadressnummer automatisch erzeugt.

3.46 LIEFERADRESSE.DELETE

Löscht einen Lieferadressdatensatz.

Parameter (obligatorisch):

- LFANR Lieferadressnummer

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Lieferadressdatensatz gelöscht werden kann.

3.47 ANSPRECHPARTNER.GET

Holt eine Liste von Ansprechpartnerdatensätzen.

Parameter (optional):

- ADRNR
- VON_ADRNR
- BIS_ADRNR schränkt die Liste auf den angegebenen Adressnummernbereich ein
- ANPNR
- VON_ANPNR
- BIS_ANPNR schränkt die Liste auf den angegebenen
 Ansprechpartnernummernbereich ein.

Verwendbare Standardparameter (optional):

- FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder
 (z.B. FELDER=ANP_1_8,ANP_9_8)
- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen
 Datensätze
- NUR_GROESSE 1=Liefert als Antwortpaket nur die Größe des in Bytes des
 tatsächlichen Antwortpakets
- SUCHE_VOLLTEXT übergibt einen Volltext-Suchbegriff. Die
 Volltextsuche wird als erster Schritt durchgeführt, danach werden
 weitere Selektionsparameter angewandt.
- FREISELEKT übergibt einen freien Selektionsausdruck.
 Bsp.: (ANP_1120_20 = '+49170123456789')
- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n)
 oder "AUTO" für automatische Ermittlung aus dem
 Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX
 Startwert für Index
- FREISELEKT_BIS_INDEX
 Endwert für Index

- FREISORT übergibt eine Komma-getrennte Liste der gewünschten Sortierfelder
FREISORT = <Feldname>[ASC|DESC][,...]
(z.B. FREISORT=ART_178_9 DESC,ART_1_25 ASC)
Es können derzeit maximal fünf Felder angegeben werden.
- MIT_LANGTEXT übergibt Langtext-Parameter. Folgender Wert ist möglich: NOTIZTEXT
- OHNE_LEERFELDER 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett ausgelassen.

Rückgabe:

```
ANSPRECHPARTNERLISTE
{
  ANSPRECHPARTNER
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

Rückgabe (NUR_ANZAHL):

```
{
  ANZAHL: <Anzahl der Datensätze>
}
```

Rückgabe (NUR_GROESSE):

```
{
  BYTES: <Größe in Bytes>
}
```

3.48 ANSPRECHPARTNER.PUT

Aktualisiert einen bestehenden Ansprechpartnerdatensatz

Parameter (obligatorisch):

- ADRNR Adressnummer
- ANPNR Ansprechpartnernummer

Parameter (optional):

- ANP_... zu setzende Felder
- OHNE_STAMMKALK 1=keine Stammdatenkalkulation durchführen
- Name eines Langtextfeldes gültige Namen siehe die GET-Funktion

Rückgabe:

Keine

3.49 ANSPRECHPARTNER.INSERT

Erstellt einen neuen Ansprechpartnerdatensatz.

Parameter (obligatorisch):

- ADRNR Adressnummer

Parameter (optional):

- ANP_... zu setzende Felder
- ANPNR Vorgabe für Ansprechpartnernummer
- OHNE_STAMMKALK 1 = Keine Stammdatenkalkulation durchführen
- Name eines Langtextfeldes gültige Namen siehe die GET-Funktion

Rückgabe:

```
{  
    ANPNR: <Adressnummer des neu erstellen Datensatzes>  
}
```

Hinweis: wird der Parameter ANPNR nicht mitgeliefert, wird eine neue Ansprechpartnernummer automatisch erzeugt.

3.50 ANSPRECHPARTNER.DELETE

Löscht einen Ansprechpartnerdatensatz.

Parameter (obligatorisch):

- ADRNR Adressnummer
- ANPNR Ansprechpartnernummer

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Ansprechpartnerdatensatz gelöscht werden kann.

3.51 VERTRETER.GET

Holt eine Liste von Vertreterdatensätzen.

Parameter (optional):

- VTRNR
- VON_VTRNR
- BIS_VTRNR schränkt die Liste auf den angegebenen Vertreternummernbereich ein

Verwendbare Standardparameter (optional):

- FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder
(z.B. FELDER=ANP_1_8,ANP_9_8)
- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen
 Datensätze
- NUR_GROESSE 1=Liefert als Antwortpaket nur die Größe des in Bytes des
 tatsächlichen Antwortpakets
- FREISELEKT übergibt einen freien Selektionsausdruck.
- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n)
 oder "AUTO" für automatische Ermittlung aus dem
 Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX
 Startwert für Index
- FREISELEKT_BIS_INDEX
 Endwert für Index
- FREISORT übergibt eine Komma-getrennte Liste der
 gewünschten Sortierfelder
 FREISORT = <Feldname>[ASC|DESC][,...]
 (z.B. FREISORT=ART_178_9 DESC,ART_1_25 ASC)
 Es können derzeit maximal fünf Felder angegeben werden.
- MIT_LANGTEXT übergibt Langtext-Parameter. Folgender Wert
 ist möglich: NOTIZTEXT
- OHNE_LEERFELDER 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett
 ausgelassen.

Rückgabe:

VERTRETERLISTE

```
{
  VERTRETER
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

Rückgabe (NUR_ANZAHL):

```
{
  ANZAHL: <Anzahl der Datensätze>
}
```

Rückgabe (NUR_GROESSE):

```
{
  BYTES: <Größe in Bytes>
}
```

3.52 VERTRETER.PUT

Aktualisiert einen bestehenden Vertreterdatensatz

Parameter (obligatorisch):

- VTRNR Vertreternummer

Parameter (optional):

- VTR_... zu setzende Felder
- OHNE_STAMMKALK 1=keine Stammdatenkalkulation durchführen
- Name eines Langtextfeldes gültige Namen siehe die GET-Funktion

Rückgabe:

Keine

Parameter (obligatorisch):

- Parameter (optional):

- Rückgabe:

Hinweis: wird der Parameter VTRNR nicht mitgeliefert, wird eine neue Vertreternummer automatisch erzeugt.

3.54 VERTRETER.DELETE

Löscht einen Vertreterdatensatz.

Parameter (obligatorisch):

- VTRNR Vertreternummer

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Vertreterdatensatz gelöscht werden kann.

3.55 TERMIN.GET

Holt eine Liste von Terminen.

Parameter (optional):

- PK holt den Termin mit dem angegebenen Primärschlüssel
- PERSNR
- VON_PERSNR
- BIS_PERSNR schränkt die Liste auf den angegebenen Personalnummernbereich ein
- DATUM
- VON_DATUM
- BIS_DATUM schränkt die Liste auf den angegebenen Datumsbereich ein.

Verwendbare Standardparameter (optional):

- FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder (z.B. FELDER=ANP_1_8,ANP_9_8)
- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen Datensätze
- NUR_GROESSE 1=Liefert als Antwortpaket nur die Größe des in Bytes des tatsächlichen Antwortpakets
- SUCHE_VOLLTEXT übergibt einen Volltext-Suchbegriff. Die Volltextsuche wird als erster Schritt durchgeführt, danach werden weitere Selektionsparameter angewandt.
- FREISELEKT übergibt einen freien Selektionsausdruck.
Bsp.: (TRM_51_8 = 10000)
- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n) oder "AUTO" für automatische Ermittlung aus dem Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX Startwert für Index
- FREISELEKT_BIS_INDEX Endwert für Index

- FREISORT übergibt eine Komma-getrennte Liste der gewünschten Sortierfelder
FREISORT = <Feldname>[ASC|DESC][,...]
(z.B. FREISORT=ART_178_9 DESC,ART_1_25 ASC)
Es können derzeit maximal fünf Felder angegeben werden.
- MIT_LANGTEXT übergibt Langtext-Parameter. Dies ist eine Komma-getrennte Liste folgender möglicher Werte: TEXT und ZUSATZTEXT
- OHNE_LEERFELDER 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett ausgelassen.
- GANZTAG 1=Anfrage bezieht sich auf Ganztagsstermine

Rückgabe:

```

TERMINLISTE
{
  TERMIN
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}

```

Rückgabe (NUR_ANZAHL):

```

{
  ANZAHL: <Anzahl der Datensätze>
}

```

Rückgabe (NUR_GROESSE):

```

{
  BYTES: <Größe in Bytes>
}

```

3.56 TERMIN.PUT

Aktualisiert einen bestehenden Termin.

Parameter (obligatorisch):

- PK Primärschlüssel

Parameter (optional):

- TRM_.. zu setzende Felder
- OHNE_STAMMKALK 1=keine Stammdatenkalkulation durchführen
- GANZTAG 1=Anfrage bezieht sich auf Ganztagestermine

Rückgabe:

Keine

3.57 TERMIN.INSERT

Erstellt einen neuen Termin.

Parameter (obligatorisch):

- PERSNR Personalnummer
- VON_DATUM Startdatum
- TITEL Titel des Termins

Parameter (optional):

- TRM_... zu setzende Felder
- BIS_DATUM Enddatum
- VON_ZEIT Startzeit
- BIS_ZEIT Endezeit
- TEXT Termin-Beschreibung
- OHNE_STAMMKALK 1 = Keine Stammdatenkalkulation durchführen

Rückgabe:

```
{
    PK: <Primärschlüssel des neu erstellen Datensatzes>
}
```

3.58 TERMIN.DELETE

Löscht einen Termin.

Parameter (obligatorisch):

- PK Primärschlüssel
- GANZTAG 1=Anfrage bezieht sich auf Ganztagestermine

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Termin gelöscht werden kann.

3.59 GESPRAECH.GET

Holt eine Liste von Gesprächen.

Parameter (optional):

- PK holt das Gespräch mit dem angegebenen Primärschlüssel
- VON_PK
- BIS_PK schränkt die Liste auf den angegebenen Schlüsselbereich ein.

Verwendbare Standardparameter (optional):

- FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder (z.B. FELDER=GSP_1_8,GSP_62_30)
- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen Datensätze
- NUR_GROESSE 1=Liefert als Antwortpaket nur die Größe des in Bytes des tatsächlichen Antwortpakets
- FREISELEKT übergibt einen freien Selektionsausdruck.
Bsp.: (GSP_1_8 = 10000)
- FREISORT übergibt eine Komma-getrennte Liste der gewünschten Sortierfelder
FREISORT = <Feldname>[ASC|DESC][,...]
(z.B. FREISORT=GSP_1_8 ASC,GSP_9_10 ASC)
Es können derzeit maximal fünf Felder angegeben werden.
- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n) oder "AUTO" für automatische Ermittlung aus dem Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX Startwert für Index
- FREISELEKT_BIS_INDEX Endwert für Index
- MIT_LANGTEXT übergibt Langtext-Parameter. Dies ist eine Komma-getrennte Liste folgender möglicher Werte: NOTIZTEXT
- OHNE_LEERFELDER 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett ausgelassen.

Rückgabe:

```

GESPRAECHELISTE
{
  GESPRAECH
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}

```

Rückgabe (NUR_ANZAHL):

```

{
  ANZAHL: <Anzahl der Datensätze>
}

```

Rückgabe (NUR_GROESSE):

```

{
  BYTES: <Größe in Bytes>
}

```

3.60 GESPRAECH.PUT

Aktualisiert ein bestehendes Gespräch.

Parameter (obligatorisch):

- PK Primärschlüssel

Parameter (optional):

- GSP_... zu setzende Felder

Rückgabe:

Keine

3.61 GESPRAECH.INSERT

Erstellt ein neues Gespräch.

Parameter (obligatorisch):

- ADRNR Adressnummer
- DATUM Startdatum
- STARTZEIT Startzeit

Parameter (optional):

- GSP_... zu setzende Felder
- ENDZEIT Endzeit
- ANPNR Ansprechpartnernummer
- PRJNR Projektnummer
- BEDNR Bedienernummer

Rückgabe:

```
{
    GSPNDX: <Primärschlüssel des neu erstellen Datensatzes>
}
```

3.62 GESPRAECH.DELETE

Löscht ein Gespräch.

Parameter (obligatorisch):

- PK Primärschlüssel

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob das Gespräch gelöscht werden kann.

Parameter (optional):

- Verwendbare Standardparameter (optional):

- 104

(z.B. FREISORT=WVL_50_10 ASC, WVL_0_10 ASC)

Es können derzeit maximal fünf Felder angegeben werden.

- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n) oder "AUTO" für automatische Ermittlung aus dem Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX Startwert für Index
- FREISELEKT_BIS_INDEX Endwert für Index
- MIT_LANGTEXT übergibt Langtext-Parameter. Dies ist eine Komma-getrennte Liste folgender möglicher Werte: NOTIZTEXT
- OHNE_LEERFELDER 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett ausgelassen.

Rückgabe:

```
WIEDERVORLAGELISTE
{
  WIEDERVORLAGE
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}
```

Rückgabe (NUR_ANZAHL):

```
{
  ANZAHL: <Anzahl der Datensätze>
}
```

Rückgabe (NUR_GROESSE):

```
{
  BYTES: <Größe in Bytes>
}
```

3.64 WIEDERVORLAGE.PUT

Aktualisiert eine bestehende Wiedervorlage.

Parameter (obligatorisch):

- WVLID Wiedervorlage-ID

Parameter (optional):

- WVL_... zu setzende Felder

Rückgabe:

Keine

3.65 WIEDERVORLAGE.INSERT

Erstellt ein neues Gespräch.

Parameter (obligatorisch):

- DATUM Datum (WVL_50_10)
- FUER_BDNR Für Bediener (WVL_65_3)

Parameter (optional):

- WVL_... zu setzende Felder

Rückgabe:

```
{  
  WVLID: <Primärschlüssel des neu erstellen Datensatzes>  
}
```

3.66 WIEDERVORLAGE.DELETE

Löscht eine Wiedervorlage.

Parameter (obligatorisch):

- WVLID Wiedervorlage-ID

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Datensatz gelöscht werden kann.

3.67 WARENGRUPPE.GET

Holt eine Liste von Warengruppen.

Parameter (optional):

- **WGRNR** holt die Warengruppe mit der angegebenen Warengruppennummer.
- **VON_WGRNR**
- **BIS_WGRNR** schränkt die Liste auf den angegebenen Nummernbereich ein.

Verwendbare Standardparameter (optional):

- **FELDER** übergibt eine Komma-getrennte Liste der gewünschten Datenfelder (z.B. FELDER=WGR_1_5,WGR_36_30)
- **NUR_ANZAHL** 1=Liefert als Antwortpaket nur die Anzahl der gefundenen Datensätze
- **NUR_GROESSE** 1=Liefert als Antwortpaket nur die Größe des in Bytes des tatsächlichen Antwortpakets
- **FREISELEKT** übergibt einen freien Selektionsausdruck.
Bsp.: (WGR_1_5 = ,WGR01')
- **FREISORT** übergibt eine Komma-getrennte Liste der gewünschten Sortierfelder
FREISORT = <Feldname>[ASC|DESC][,...]
(z.B. FREISORT=WGR_1_5 ASC)
Es können derzeit maximal fünf Felder angegeben werden.
- **FREISELEKT_KEY** Nummer des gewünschten Schlüssels (0...n) oder "AUTO" für automatische Ermittlung aus dem Selektionsausdruck (experimentell!)
- **FREISELEKT_VON_INDEX**
Startwert für Index
- **FREISELEKT_BIS_INDEX**
Endwert für Index
- **MIT_LANGTEXT** übergibt Langtext-Parameter. Dies ist eine Komma-getrennte Liste folgender möglicher Werte: NOTIZTEXT, WARNTTEXT
- **OHNE_LEERFELDER** 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett ausgelassen.

Rückgabe:

```

WARENGRUPPENLISTE
{
  WARENGRUPPE
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}

```

Rückgabe (NUR_ANZAHL):

```

{
  ANZAHL: <Anzahl der Datensätze>
}

```

Rückgabe (NUR_GROESSE):

```

{
  BYTES: <Größe in Bytes>
}

```

3.68 WARENGRUPPE.PUT

Aktualisiert eine bestehende Warengruppe.

Parameter (obligatorisch):

- WGRNR Warengruppennummer

Parameter (optional):

- WGR_... zu setzende Felder

Rückgabe:

Keine

3.69 WARENGRUPPE.INSERT

Erstellt eine neue Warengruppe.

Parameter (obligatorisch):

- WGRNR Warengruppennummer

Parameter (optional):

- WGR_... zu setzende Felder

Rückgabe:

```
{  
  WGRNR: <Primärschlüssel des neu erstellen Datensatzes>  
}
```


3.70 WARENGRUPPE.DELETE

Löscht eine Warengruppe.

Parameter (obligatorisch):

- WGRNR Warengruppennummer

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob die Warengruppe gelöscht werden kann.


```

{
  LAGER
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}

```

Rückgabe (NUR_ANZAHL):

```

{
  ANZAHL: <Anzahl der Datensätze>
}

```

Rückgabe (NUR_GROESSE):

```

{
  BYTES: <Größe in Bytes>
}

```

3.72 LAGER.PUT

Aktualisiert einen bestehenden Lagerstammsatz

Parameter (obligatorisch):

- LAGER Lagerident

Parameter (optional):

- LAG_... zu setzende Felder

Rückgabe:

Keine

3.73 LAGER.INSERT

Erstellt einen neuen Lagerstammsatz.

Parameter (obligatorisch):

- LAGER Lagerident

Parameter (optional):

- LAG_... zu setzende Felder

Rückgabe:

```
{  
  LAGID: <Primärschlüssel des neu erstellen Datensatzes>  
}
```

3.74 LAGER.DELETE

Löscht einen Lagerstammsatz.

Parameter (obligatorisch):

- LAGER Lagerident

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Lagerstammsatz gelöscht werden kann.

3.75 MPKATALOG.GET

Holt die Marketplace-Katalogliste, wahlweise mit Kategorien und zugehörigen Artikeln.

Parameter (optional):

- KATALOG
- VON_KATALOG
- BIS_KATALOG schränkt die Liste auf den angegebenen Katalognummernbereich ein
- MIT_KATEGORIEN 1=Ruft für jeden Katalog auch den Kategorienbaum ab
- KATEGORIE_FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder für die Kategorien
- MIT_KATEGORIEN_LANGTEXT
übergibt Langtext-Parameter. Dies ist eine Komma-getrennte Liste folgender möglicher Werte: KURZ_n, LANG_n, META_n, KEYS_n, wobei n = 0, .., 4 die Sprach-ID ist.
- MIT_ARTIKEL 1=Ruft für jeden Katalog auch die Liste der enthaltenen Artikel ab. Wenn MIT_KATEGORIEN=1, werden die Artikel pro Kategorie, nicht pro Katalog, abgerufen.
- ARTIKEL_FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder für die Artikel
- MIT_LAGERBESTAND 1=Ruft Lagerbestände in die Fehler ART_759_10 und ART_769_10 ab
- LAGER
- VON_LAGER
- BIS_LAGER schränkt Lager für Lagerbestandsermittlung auf den angegebenen Bereich ein.

Verwendbare Standardparameter (optional):

- FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder (z.B. FELDER=KAT_0_5,KAT_5_60)
- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen Datensätze

- **NUR_GROESSE** 1=Liefert als Antwortpaket nur die Größe des in Bytes des tatsächlichen Antwortpakets
- **FREISELEKT** übergibt einen freien Selektionsausdruck.
Bsp.: (KAT_0_5 = 12345)
- **OHNE_LEERFELDER** 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett ausgelassen.

Rückgabe (Beispiel, MIT_KATEGORIEN=1 und MIT_ARTIKEL=1):

KATALOGLISTE:

```
{
  KATALOG:
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste Katalog oder alle Felder
      KATEGORIENLISTE:
      {
        KATEGORIE:
        [
          {
            SNR: <Satznummer>
            Felder gemäß Feldliste Kategorien oder alle Felder
            KATEGORIENLISTE:
            {
              KATEGORIE:
              [
                ...
              ],
              ARTIKELLISTE:
              {
                ARTIKEL:
                [
                  {
                    SNR: <Satznummer>
                    Felder gemäß Feldliste
                    Artikel
                  }
                  ...
                ]
                ANZAHL: <Anzahl Artikel>
              }
            }
            ...
          ]
          ANZAHL: <Anzahl Kategorien>
        }
      }
    }
  ]
}
```



```
]
ANZAHL: <Anzahl Kataloge>
}
```

3.76 MPKATEGORIE.GET

Holt eine Liste von Kategorien.

Parameter (optional):

- **KTGNR** holt die Kategorie mit der angegebenen Kategorienummer.
- **VON_KTGNR**
- **BIS_KTGNR** schränkt die Liste auf den angegebenen Nummernbereich ein.

Verwendbare Standardparameter (optional):

- **FELDER** übergibt eine Komma-getrennte Liste der gewünschten Datenfelder
(z.B. FELDER=KTG_0_10, KTG_10_10)
- **NUR_ANZAHL** 1=Liefert als Antwortpaket nur die Anzahl der gefundenen
Datensätze
- **NUR_GROESSE** 1=Liefert als Antwortpaket nur die Größe des in Bytes des
tatsächlichen Antwortpakets
- FREISELEKT übergibt einen freien Selektionsausdruck.
Bsp.: (KTG_0_10 >= 10)
- FREISORT übergibt eine Komma-getrennte Liste der
gewünschten Sortierfelder
FREISORT = <Feldname>[ASC|DESC][,...]
(z.B. FREISORT=KTG_0_10 ASC)
Es können derzeit maximal fünf Felder angegeben werden.
- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n)
oder "AUTO" für automatische Ermittlung aus dem
Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX
Startwert für Index
- FREISELEKT_BIS_INDEX
Endwert für Index
- **OHNE_LEERFELDER** 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett
ausgelassen.

Rückgabe:

KATEGORIENLISTE

```

{
  KATEGORIE
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}

```

Rückgabe (NUR_ANZAHL):

```

{
  ANZAHL: <Anzahl der Datensätze>
}

```

Rückgabe (NUR_GROESSE):

```

{
  BYTES: <Größe in Bytes>
}

```

3.77 MPKATEGORIE.PUT

Aktualisiert eine bestehende Kategorie.

Parameter (obligatorisch):

- KTGNR Kategorienummer

Parameter (optional):

- KTG_... zu setzende Felder

Rückgabe:

Keine

3.78 MPKATEGORIE.INSERT

Erstellt eine neue Kategorie.

Parameter (obligatorisch):

- KTGNR Kategorienummer

Parameter (optional):

- KTG_... zu setzende Felder

Rückgabe:

```
{
    KTGNR: <Primärschlüssel des neu erstellen Datensatzes>
}
```

3.79 MPKATEGORIE.DELETE

Löscht eine Kategorie.

Parameter (obligatorisch):

- KTGNR Kategorienummer

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob die Kategorie gelöscht werden kann.

3.80 EANCODE.GET

Holt eine Liste von EAN-Codes

Parameter (optional):

- EAN holt den EAN-Code mit der angegebenen ID
- VON_EAN
- BIS_EAN schränkt die Liste auf den angegebenen ID-Bereich ein.
- VON_ARTNR
- BIS_ARTNR schränkt die Liste auf den angegebenen Artikelnummernbereich ein.

Verwendbare Standardparameter (optional):

- FELDER übergibt eine Komma-getrennte Liste der gewünschten Datenfelder
(z.B. FELDER=EAN_19_25,EAN_76_25)
- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen
Datensätze
- NUR_GROESSE 1=Liefert als Antwortpaket nur die Größe des in Bytes des
tatsächlichen Antwortpakets
- FREISELEKT übergibt einen freien Selektionsausdruck.
Bsp.: (EAN_19_25 >= ,EAN123')
- FREISORT übergibt eine Komma-getrennte Liste der
gewünschten Sortierfelder
FREISORT = <Feldname>[ASC|DESC][,...]
(z.B. FREISORT=EAN_71_25 ASC,EAN_19_25)
Es können derzeit maximal fünf Felder angegeben werden.
- FREISELEKT_KEY Nummer des gewünschten Schlüssels (0...n)
oder "AUTO" für automatische Ermittlung aus dem
Selektionsausdruck (experimentell!)
- FREISELEKT_VON_INDEX
Startwert für Index
- FREISELEKT_BIS_INDEX
Endwert für Index
- OHNE_LEERFELDER 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett
ausgelassen.

Rückgabe:

```

EANCODELISTE
{
  EANCODE
  [
    {
      SNR: <Satznummer>
      Felder gemäß Feldliste oder alle Felder
    },
    ...
  ]
  ANZAHL: <Anzahl gelieferter Datensätze>
}

```

Rückgabe (NUR_ANZAHL):

```

{
  ANZAHL: <Anzahl der Datensätze>
}

```

Rückgabe (NUR_GROESSE):

```

{
  BYTES: <Größe in Bytes>
}

```


3.81 EANCODE.PUT

Aktualisiert einen bestehenden EAN-Codesatz

Parameter (obligatorisch):

- EAN ID

Parameter (optional):

- EAN_... zu setzende Felder

Rückgabe:

Keine

3.82 EANCODE.INSERT

Erstellt einen neuen EAN-Codesatz.

Parameter (obligatorisch):

- EAN ID

Parameter (optional):

- EAN_... zu setzende Felder

Rückgabe:

```
{  
  EANCODE: <Primärschlüssel des neu erstellen Datensatzes>  
}
```

3.83 EANCODE.DELETE

Löscht einen EAN-Codesatz.

Parameter (obligatorisch):

- EAN ID

Rückgabe:

Keine

Hinweis:

Es wird eine Prüfung durchgeführt, ob der Datensatz gelöscht werden kann.

3.84 VERSANDART.GET

Holt eine Liste von Versandarten.

Parameter (optional):

Parameter	Beschreibung
VANR	
VON_VANR	
BIS_VANR	Schränkt die Liste auf den angegebenen Bereich ein.

Verwendbare Standardparameter (optional):

Parameter	Erlaubte Werte/Anmerkung
FELDER	
NUR_ANZAHL	
NUR_GROESSE	
FREISELEKT	
FREISORT	
OHNE_LEERFELDER	

Rückgabe:

```
VERSANDARTLISTE:
{
  VERSANDART:
  [
    SNR: <Satznummer>
    Felder gemäß Feldliste oder alle Felder
```

```
] , ...  
  ANZAHL: <Anzahl gelieferter Datensätze>  
}
```

3.85 VERSANDART.PUT

Aktualisiert einen bestehenden EAN-Codesatz

Parameter (obligatorisch):

Parameter	Beschreibung
VANR	Versandarten-Nummer
FVE_...	Zu setzende Felder

Rückgabe:

Keine

3.86 VERSANDART.INSERT

Fügt einen neuen Versandarten-Satz.

Parameter (obligatorisch):

Parameter	Beschreibung
VANR	Versandarten-Nummer
FVE_...	Zu setzende Felder

Rückgabe:

Keine

3.87 VERSANDART.DELETE

Löscht einen Versandarten-Satz.

Parameter (obligatorisch):

Parameter	Beschreibung
VANR	Versandarten-Nummer

Rückgabe:

Keine

3.88 MEDIEN.GET

Liefert eine Mediendatei als binären Download oder als BASE64-codierten Text.

Parameter (obligatorisch):

- ID eine Medien-ID

Header-Parameter:

- WWSVC-ACCEPT-RESULT-TYPE
JSON=BASE64-codierter Text
BIN=binärer Download

Rückgabe:

Abhängig von WWSVC-ACCEPT-RESULT-TYPE, die Mediendatei als binärer Download oder als JSON-Struktur wie folgt:

```
"MEDIENLISTE": {  
  "MEDIEN": [  
    {  
      "MIMETYPE": <Mime-Typ, z.B. image/bmp>  
      "DATA": BASE64-codierter Dateiinhalt  
    }  
  ]  
}
```

Anstelle von WWSVC-ACCEPT-RESULT-TYPE im HTTP-Header kann auch der Funktions-parameter RESULT_TYPE mit dem Wert „JSON“ bzw. „BIN“ mitgegeben werden.

3.89 FILE.GET

Liefert eine Datei als binären Download oder als BASE64-codierten Text.

Parameter (obligatorisch):

Parameter	Beschreibung
PFAD	Pfad zur Datei

Parameter (optional):

Parameter	Beschreibung
GET_DIRECTORY	1: holt den Verzeichnisinhalt; in diesem Fall ist PFAD der Ordnerpfad, dessen Verzeichnis gelesen werden soll.
PATTERN	Suchmuster für GET_DIRECTORY, z.B. „*.txt“.
OFFLINE_KASSE	1: spezieller Modus für die Offline-Kasse

Header-Parameter:

Parameter	Beschreibung
WWSVC-ACCEPT-RESULT-TYPE	JSON=BASE64-codierter Text BIN=binärer Download

Rückgabe:

Abhängig von WWSVC-ACCEPT-RESULT-TYPE, die Mediendatei als binärer Download oder als JSON-Struktur wie folgt:

```
FILELISTE:
```

```

{
    FILE:
    [
        {
            MIMETYPE: <Mime-Typ, z.B. image/bmp>
            DATA: BASE64-codierter Dateiinhalt
        }
    ]
}

```

Rückgabe (GET DIRECTORY=1):

```

FILELISTE:
{
    FILE:
    [
        NAME: <Dateiname>
        GROESSE: <Größe in Bytes>
        CTIME: <Datum/Uhrzeit der Erstellung>
        MTIME: <Datum/Uhrzeit der letzten Änderung>
        DIRECTORY: <0 = reguläre Datei, 1 = Unterverzeichnis>
    ], ...
}

```

Hinweise:

Aus Sicherheitsgründen wird PFAD immer relativ zum aktiven Mandantenpfad interpretiert. Es ist nicht möglich, absolute Pfade anzugeben, außerdem wird auf Konstruktionen mit „...“ geprüft, um sicherzustellen, dass der Mandantenpfad nicht verlassen werden kann. Es ist zudem nicht möglich, Dateien mit den Endungen „.DAT“, „.KEY“ und „.DTK“ herunterzuladen.

Falls OFFLINE_KASSE=1, ist der Basispfad nicht mehr der Mandantenpfad, sondern der Ordner „OFFLINE-KASSE“ im WW-Applikationsordner.

3.90 FILE.PUT

Lädt eine Datei auf den WEBWARE-Server hoch.

Der Aufruf muss über EXECURL erfolgen und die hochzuladende Datei im http-Body übertragen werden.

Parameter (optional):

Parameter	Beschreibung
PFAD	Zielpfad der Datei (inklusive Dateiname). Wird dieser Parameter nicht angegeben, wird er mit dem Dateinamen vorbelegt.
UEBERSCHREIBEN	1: ggf. vorhandene Datei gleichen Namens überschreiben.
OFFLINE_KASSE	1: spezieller Modus für die Offline-Kasse

Rückgabe:

Keine

Hinweise:

Aus Sicherheitsgründen wird PFAD immer relativ zum aktiven Mandantenpfad interpretiert. Es ist nicht möglich, absolute Pfade anzugeben, außerdem wird auf Konstruktionen mit „..“ geprüft, um sicherzustellen, dass der Mandantenpfad nicht verlassen werden kann. Es ist zudem nicht möglich, Dateien mit den Endungen „.DAT“, „.KEY“ und „.DTK“ hochzuladen.

Falls OFFLINE_KASSE=1, ist der Basispfad nicht mehr der Mandantenpfad, sondern der Ordner „OFFLINE-KASSE“ im WW-Applikationsordner.

3.91 FORMULAR.GET

Holt eine Liste von Formularen.

Parameter (optional):

Parameter	Beschreibung
BEREICH	Formularbereich <ul style="list-style-type: none">• FMTOP AUS• FMTFA AUS• FMTFABEL• FMTFIAUS• FMTFIMAH• FMTFIZVK• FMTANAUS• FMTKRAUS• FMTFASRV• FMTLOAUS
VON_FMTNR	
BIS_FMTNR	Schränkt die Liste auf den angegebenen Bereich ein. Gültige Werte: 0 – 999 bzw. 0 – 9999.
NAME	Name eines Formulars

Hinweis:

Es muss zwingend entweder NAME oder BEREICH angegeben werden. VON_FMTNR und BIS_FMTNR sind optional.

Rückgabe (Beispiel):

```

"FORMULARLISTE": {
  "FORMULAR": [
    {
      "FMTNR": "000",
      "NAME": "FMTFAAUS.000",
      "BEZEICHNUNG": "",
      "PFAD": "FMT\\FMTFAAUS.000",
      "LISTENBASIS": "ADRESSDATEN",
      "SELEKTIONSPARAMETERLISTE": {
        "SELEKTIONSPARAMETER": [
          {
            "VARIABLE": "PE_VON_NR",
            "ANZEIGENAME": "Von Adressnummer",
            "VORBESETZUNG": "",
            "LAENGE": "8",
            "EINGABEART": "R0"
          },
          {
            "VARIABLE": "PE_BIS_NR",
            "ANZEIGENAME": "Bis Adressnummer",
            "VORBESETZUNG": "99999999",
            "LAENGE": "8",
            "EINGABEART": "R0"
          },
          ...
        ],
        "ANZAHL": ...
      },
      "EINGABEPARAMETERLISTE": {
        "EINGABEPARAMETER": [
          {
            "VARIABLE": "IE1",
            "ANZEIGENAME": "Mit Zeilenschattierung",
            "VORBESETZUNG": "J",
            "LAENGE": "1",
            "EINGABEART": "AJN"
          },
          {
            "VARIABLE": "IE2",
            "ANZEIGENAME": "Fax und Mobiltelefon drucken",
            "VORBESETZUNG": "0",
            "LAENGE": "1",
            "EINGABEART": "B01"
          }
        ],
        "ANZAHL": "2"
      }
    }
  ]
}

```

3.92 FORMULAR.EXEC

Startet den Ausdruck eines Formulars.

Parameter (obligatorisch):

Parameter	Beschreibung
NAME	Name eines Formulars
DRUCKER	Name eines Druckers

Parameter (optional):

Parameter	Beschreibung
IE...	Eingabevariablen des Formulars (IE1, IE2, etc.)
Name einer Selektionsvariablen	Selektionsvariablen des Formulars (PE_VON_NR, PE_BIS_NR, etc.)

Rückgabe:

Keine

3.93 BEDIENER.GET

Ruft die Bedienerliste ab.

Parameter (optional):

- BDNR
- VON_BDNR
- BIS_BDNR schränkt die Liste auf den angegebenen Bedienernummernbereich ein
- MIT_MODULBERECHTIGUNGEN
1=es wird pro Bediener die Liste der Modulberechtigungen ausgegeben.

Verwendbare Standardparameter (optional):

- NUR_ANZAHL 1=Liefert als Antwortpaket nur die Anzahl der gefundenen Datensätze
- NUR_GROESSE 1=Liefert als Antwortpaket nur die Größe des in Bytes des tatsächlichen Antwortpakets
- OHNE_LEERFELDER 1=Datenfelder ohne Inhalt werden im Ergebnissatz komplett ausgelassen.

Rückgabe:

```
"BEDIENERLISTE": {
  "BEDIENER": [
    {
      "BDNR": "000",
      "PERSONAL_NR": "1",
      "NAME": "Systemverwalter",
      "KUERZEL": "",
      "ABTEILUNG": "",
      "BEMERKUNG_1": "",
      "BEMERKUNG_2": "",
      "BEMERKUNG_3": "",
      "BEMERKUNG_4": "",
      "BEMERKUNG_5": "",
      "TELEFON": "",
      "GESPERRT": "N",
      "ANWENDUNGSKONTO": "BüroWARE/WEBWARE",
      "BEDIENERART": "Systemverwalter/Designer",
      "MODULLISTE": {
        "MODUL": [
          {
```

```
"NAME": "Warenwirtschaft",  
"ERLAUBT": "J"  
},  
{  
  "NAME": "Verkaufsförderung",  
  "ERLAUBT": "N"  
},  
...
```


3.94 GET_RELATION.EXEC

Führt eine GET_RELATION aus.

Parameter (obligatorisch):

- NR Die Nummer der GET_RELATION

Parameter (optional)

- P1..Pn weitere Parameter der GET_RELATION

Anmerkung: Die Parameter P1 bis Pn können in beliebiger Reihenfolge angegeben werden. Die Nummer entspricht dabei der Position des Parameters in dem vergleichbaren Aufruf der GET_RELATION in einem Eventskript. Soll ein Parameter leer bleiben, muss er mit leerem Inhalt übergeben werden.

Rückgabe:

```
{  
    GET_RESULT: <Rückgabewert der GET_RELATION>  
}
```

3.95 PUT_RELATION.EXEC

Führt eine PUT_RELATION aus.

Parameter (obligatorisch):

- NR Die Nummer der PUT_RELATION

Parameter (optional)

- P1..Pn weitere Parameter der PUT_RELATION
- ALS_PUTADD_AUSFUEHREN 1 = PUT wird als PUT_ADD ausgeführt.

Anmerkung: Die Parameter P1 bis Pn können in beliebiger Reihenfolge angegeben werden. Die Nummer entspricht dabei der Position des Parameters in dem vergleichbaren Aufruf der PUT_RELATION in einem Eventskript. Soll ein Parameter leer bleiben, muss er mit leerem Inhalt übergeben werden.

Rückgabe:

Keine

3.96 WORKFLOW.EXEC

Führt ein Workflowskript aus.

Parameter (obligatorisch):

- ID Die ID des Workflow-Skriptes

Parameter (optional)

- MEM_... MEM-Felder, deren Werte vor der Skriptausführung gesetzt werden. Diese können zur Übergabe von Parameter genutzt werden.
- RESULT_VAR Name eines MEM-Feldes, das zur Rückgabe eines Wertes genutzt wird.

Anmerkung: Es muss sichergestellt sein, dass das Workflowskript nicht-interaktiv ist, d.h. das Skript darf keine Bedienerinteraktion erfordern, da der Serviceprozess in diesem Fall hängen bleibt.

Rückgabe:

```
{  
    RESULT: <Wert des in RESULT_VAR angegebenen MEM-Feldes>  
}
```

3.97 STATARTLG.EXEC

Führt die Statistikfunktion STATARTLG aus.

Parameter (obligatorisch):

- ARTNR Artikelnummer
- ART
 - 0 = Lagerbestand
 - 1 = Lagerbestand kalkuliert
 - 2 = Inventur DM Wert (obsolet)
 - 3 = Inventur Mengen
 - 4 = Bestand Reparaturlager
 - 5 = In Bearbeitung/Reparatur im Hause
 - 6 = Bearbeitung beim Lieferant
 - 7 = Nach Bearbeitung von Lieferant zurück
 - 8 = Lagerbestand reserviert
 - 9 = freies Statistikfeld 1
 - 10 = freies Statistikfeld 2
 - 11 = freies Statistikfeld 3
 - 12 = freies Statistikfeld 4
 - 13 = freies Statistikfeld 5
 - 14 = Menge reserviert
 - 15 = Menge disponiert reserviert
 - 16 = Menge alternativ reserviert
 - 17 = Menge alternativ disponiert reserviert
 - 18 = Inventur Euro Wert

Parameter (optional):

- VON_LAGER
- BIS_LAGER Einschränkung auf einen Lagerbereich
- ALTERNATIVE_LAGEREINHEIT
 - 0 = normale Menge
 - 1 = alternative Lagereinheit
- WILDCARD 1 = Artikelnummer mit ArtNr* berechnen
- LAGERGESAMT_IGNOREN
 - 1 = Keine Prüfung auf LAG_174_1

Rückgabe:

```
{
  RESULT: <ermittelter Statistikwert>
}
```

3.98 STATARTUM.EXEC

Führt die Statistikfunktion STATARTUM aus.

Parameter (obligatorisch):

- ARTNR Artikelnummer
- ART 0 = Wareneingang Umsatz
 1 = Wareneingang Menge
 2 = Warenausgang Umsatz
 3 = Warenausgang Menge
 4 = Rohertrag Umsatz
 5 = WA Angebot Umsatz
 6 = WA Angebot Menge
 7 = WA Auftrag Umsatz
 8 = WA Auftrag Menge
 9 = WA Lieferschein Umsatz
 10 = WA Lieferschein Menge
 11 = WE Bestellung Umsatz
 12 = WE Bestellung Menge
 13 = WE Lieferschein Umsatz
 14 = WE Lieferschein Menge
 15 = Lagereingang Umsatz
 16 = Lagereingang Menge
 17 = Lagerausgang Umsatz
 18 = Lagerausgang Menge
 19 = WE Fertigungsauftrag Menge
 20 = WE Fertigungsauftrag in Produktion
 21 = WE Fertigungsauftrag auf Lager
 22 = WE Fertigungsauftrag Wert
 23 = Reserviert
 24 = dispo_reserviert
 25 = aReserviert
 26 = dispo_aReserviert
 27 = WE Fertigungsauftrag Menge
 28 = WE Fertigungsauftrag in Produktion
 29 = WE Fertigungsauftrag auf Lager
 30 = WE Fertigungsauftrag Wert

Parameter (optional):

- VON_JAHR

- BIS_JAHR Einschränkung auf Geschäftsjahresbereich (Standard: VON_JAHR = 0, BIS_JAHR = 0)
- VON_PERIODE
- BIS_PERIODE Einschränkung auf Periodenbereich (Standard: VON_PERIODE = 1, BIS_PERIODE = 12)
- DATUM Einschränkung auf Datum (übersteuert JAHR und PERIODE)
- ALTERNATIVE_LAGEREINHEIT
 0 = normale Menge
 1 = alternative Lagereinheit
- WILDCARD 1 = Artikelnummer mit ArtNr* berechnen

Rückgabe:

```
{  
    RESULT: <ermittelter Statistikwert>  
}
```

3.99 STATLGDATEXEC

Führt die Statistikfunktion STATLGDAT aus.

Parameter (obligatorisch):

- ARTNR Artikelnummer
- VON_LAGER Lager
- ART
 - 0 = Standardermittlung
 - 1 = Mit WE-Bestellungen
 - 2 = Mit WA-Aufträgen
 - 3 = Mit WE-Bestellungen und WA-Aufträgen

Parameter (optional):

- BIS_LAGER falls angegeben, erfolgt die Ermittlung für alle Läger VON_LAGER bis BIS_LAGER
- MIT_MATERIALUMLAUF
 - 0 = ohne Berücksichtigung des Materialumlaufts
 - 1 = mit Berücksichtigung des Materialumlaufts
- ALTERNATIVE_LAGEREINHEIT
 - 0 = normale Menge
 - 1 = alternative Lagereinheit
- MIT_LAGERGESAMT
 - 1 = Mit Prüfung auf LAG_174_1

Rückgabe:

```
{  
  RESULT: <ermittelter Statistikwert>  
}
```

3.100 STATADR.EXEC

Führt die Statistikfunktion STATADR aus.

Parameter (obligatorisch):

- ADRNR Adressnummer
- ART 0 = Umsatz
 1 = Rohertrag
 2 = Freie Summen (nur Nebenkosten)
- BELEGART 5 = Umsatz allgemein (Rechnungen/Gutschriften)
 0 = WA-Angebot
 1 = WA-Auftrag
 2 = WA-Lieferschein
 3 = WE-Bestellung
 4 = WE-Lieferschein
 6 – 104 = Lagerwert 01-99 (Adresslager)

Parameter (optional):

- JAHR Geschäftsjahr 0-9 (Standard: JAHR = 0)
- VON_PERIODE
- BIS_PERIODE Einschränkung auf Perioden 1-12 ((Standard:
 VON_PERIODE = 1, BIS_PERIODE = 12)

Hinweis: die Parameter JAHR, VON_PERIODE und BIS_PERIODE sind nur bei ART = 5 verfügbar.

Rückgabe:

```
{  
    RESULT: <ermittelter Statistikwert>  
}
```


3.101 STATAAR.EXEC

Führt die Statistikfunktion STATAAR aus.

Parameter (obligatorisch):

- ADRNR Adressnummer
- ARTNR Artikelnummer
- ART 0 = Umsatz
 1 = Menge
 2 = Rohertrag

Parameter (optional):

- JAHR Geschäftsjahr 0-9 (Standard: JAHR = 0)
- VON_PERIODE
- BIS_PERIODE Einschränkung auf Perioden 1-12 (Standard:
 VON_PERIODE = 1, BIS_PERIODE = 12)

Rückgabe:

```
{  
    RESULT: <ermittelter Statistikwert>  
}
```

3.102 STATPRJ.EXEC

Führt die Statistikfunktion STATPRJ aus.

Parameter (obligatorisch):

- PRJNR Projektnummer
- ART 0 = Umsatz
 1 = Rohertrag
 2 = Freie Summen
 3 = Einkauf
 4 = Bezugskosten
- BELEGART 0 = WA Angebot
 1 = WA-Auftrag
 2 = WA-Lieferschein
 3 = WE-Bestellung
 4 = WE-Lieferschein
 5 = Rechnungen/Gutschriften

Parameter (optional):

- JAHR Jahr (Standard: JAHR = 0)
- VON_PERIODE
- BIS_PERIODE Einschränkung auf Periode 1-12 (Standard: VON_PERIODE
= 1, BIS_PERIODE = 12)

Rückgabe:

```
{  
    RESULT: <ermittelter Statistikwert>  
}
```

3.103 STATLFA.EXEC

Führt die Statistikfunktion STATLFA aus.

Parameter (obligatorisch):

- LFANR Lieferadressnummer
- ART 0 = Umsatz
 1 = Rohertrag
 2 = Freie Summen
 3 = Einkauf
 4 = Bezugskosten
- BELEGART 0 = WA Angebot
 1 = WA-Auftrag
 2 = WA-Lieferschein
 3 = WE-Bestellung
 4 = WE-Lieferschein
 5 = Rechnungen/Gutschriften

Parameter (optional):

- JAHR Jahr (Standard: JAHR = 0)
- VON_PERIODE
- BIS_PERIODE Einschränkung auf Periode 1-12 (Standard: VON_PERIODE
= 1, BIS_PERIODE = 12)

Rückgabe:

```
{
  RESULT: <ermittelter Statistikwert>
}
```

3.104 STATWGR.EXEC

Führt die Statistikfunktion STATWGR aus.

Parameter (obligatorisch):

- WGRNR Warengruppennummer
- ART Falls ADRNR *nicht* angegeben wird:
0 = WE Umsatz
1 = WE Menge
2 = WA Umsatz
3 = WA Menge
4 = Rohertrag

Falls ADRNR angegeben wird:
0 = Umsatz
1 = Menge
2 = Rohertrag

Parameter (optional):

- ADRNR Adressnummer (Standard: leer)
- JAHR Jahr (Standard: JAHR = 0)
- VON_PERIODE
- BIS_PERIODE Einschränkung auf Periode 1-12 (Standard: VON_PERIODE
= 1, BIS_PERIODE = 12)

Rückgabe:

```
{  
    RESULT: <ermittelter Statistikwert>  
}
```

3.105 STATVTWGR.EXEC

Führt die Statistikfunktion STATVTWGR aus.

Parameter (obligatorisch):

- WGRNR Warengruppennummer
- ART 0 = Umsatz
 1 = Menge
 2 = Rohertrag

Parameter (optional):

- VTRNR Vertreternummer (Standard: keine Angabe = alle Vertreter)
- JAHR Jahr (Standard: JAHR = 0)
- VON_PERIODE
- BIS_PERIODE Einschränkung auf Periode 1-12 (Standard: VON_PERIODE
= 1, BIS_PERIODE = 12)

Rückgabe:

```
{  
    RESULT: <ermittelter Statistikwert>  
}
```

3.106 IMPORT.EXEC

Führt einen Datenimport aus

Der Aufruf muss über EXECURL erfolgen und die Importdatei im http-Body übertragen werden. Die Importdatei muss im Standardimportformat vorliegen.

Parameter (optional):

Parameter	Beschreibung
IMPORTMODELLNR	Nummer des gewünschten Importmodells (0 – 99). Die hoch-geladene Importdatei muss im passenden Format vorliegen.

Rückgabe:

Das Importprotokoll wie im nachfolgenden Beispiel:

```
{
  "json": {
    "COMRESULT": {
      "STATUS": 200,
      "CODE": "200 OK",
      "INFO": "Kein Fehler",
      "INFO2": "",
      "INFO3": ""
    },
    "IMPORTPROTOKOLL": [
      "IMPORTMODELL": 02,
      "ANGELEGT      Artikel      Artikel1      ArtikelmitBezeichnung1",
      "ANGELEGT      Artikel      Artikel2      ArtikelmitBezeichnung2",
      "ANGELEGT      Adresse      10010        AdresseBezeichnungA",
      "ANGELEGT      Belegkopf     A20161033  10000",
      "ANGELEGT      Belegposition A20161033  Artikel19Prozent",
      "ANGELEGT      Belegkopf     A20161034  10011",
      "ANGELEGT      Belegposition A20161034  Artikel19Prozent",
      "ANGELEGT      :           4 <- GESAMT",
      "GEÄNDERT      :           11 <- GESAMT",
      "HINZUGEFGÜGT :           0 <- GESAMT",
      "GELÖSCHT      :           0 <- GESAMT"
    ],
    "RESULT": "OK"
  }
}
```

3.107 MEMTAB.EXEC

Vorgehensweise:

MEMTABs werden im Hauptspeicher eines Prozessor-Knotens erzeugt. Daher müssen aufeinanderfolgende Zugriffe auf dieselbe MEMTAB immer auf demselben Prozessor-Knoten ausgeführt werden.

Dazu wird bei MEMTAB.EXEC mit CMD = INIT eine „Dynamisches-Objekt-ID“ (im folgenden „DynObjID“) erstellt, es sei denn, bei dem Aufruf wird bereits eine DynObjID mitgeliefert (dadurch können mehrere MEMTABs auf dem gleichen Prozessor-Knoten erzeugt werden). Das Konzept ist das Gleiche wie beim Cursor-Handle, auch die Übergabe der ID erfolgt analog über ein Header-Feld.

Sind die MEMTABs mit CMD = TABLOESCHEN wieder gelöscht, ist als letzte Aktion die Funktion WWAPP.CLOSE_DYNOBJ mit der entsprechenden DynObjID aufzurufen, damit das Handle wieder freigegeben wird.

Parameter:

Es werden die gleichen Aktionen unterstützt, die auch in Stammdatenkalkulationen bzw. als GET_RELATION für MEMTABs verfügbar sind.

- CMD = INIT
Initialisiert eine neue MEMTAB
 - TABNAME = Eindeutiger Name der Tabelle
 - P1 = ArtPIndex (Art des Primärindex (0 = alphanumerisch, 1 = numerisch max. 10-stellig))
 - P2 = Länge PIndex (Angabe der Indexlänge des Primärschlüssels bei ArtPIndex = 0)
 - P3 = Anzahl der numerischen Spalten
 - P4 = Anzahl der Datums-Spalten
 - P5 = Anzahl der alphanumerischen Spalten
- CMD = TABLOESCHEN
Löscht eine MEMTAB
 - TABNAME = Eindeutiger Name der Tabelle
- CMD = NEUEZEILE
Erzeugt eine neue Zeile in einer MEMTAB

- TABNAME = Eindeutiger Name der Tabelle
 - P1 = Primärindex
- CMD = LOESCHEPINDEX
Löscht eine Zeile anhand des Primärschlüssels
 - TABNAME = Eindeutiger Name der Tabelle
 - P1 = Primärindex
- CMD = LOESCHENUMMER
Löscht eine Zeile anhand der Zeilennummer
 - TABNAME = Eindeutiger Name der Tabelle
 - P1 = Zeilennummer
- CMD = SETZEWERT
Setzt den Wert einer Zelle
 - TABNAME = Eindeutiger Name der Tabelle
 - P1 = Primärindex der Zeile
 - P2 = Art (wie wird der neue Wert gesetzt?) (0 = überschreiben, 1 = addieren, 2 = subtrahieren)
 - P3 = Spaltenname (Name der Spalte, deren Wert gesetzt werden soll)
Ax = Alphanummerische Spalte ansprechen
Nx = Nummerische Spalte ansprechen
Dx = Datum Spalte ansprechen
x steht für die entsprechende Spaltennummer, wobei bei 0 begonnen wird
 - P4 = Wert1
 - P5 = Spaltenname 2 (optional)
 - P6 = Wert 2 (optional)
 - ...
- CMD = HOLEWERT
Liefert den Wert einer Zelle
 - TABNAME = Eindeutiger Name der Tabelle
 - P1 = PIndex (Primärindex der Zeile, für die der Spaltenwert ermittelt wird)
 - P2 = Spaltenname (Name der Spalte, deren Wert ermittelt werden soll)
Ax = Alphanummerische Spalte ansprechen
Nx = Nummerische Spalte ansprechen
Dx = Datum Spalte ansprechen
x steht für die entsprechende Spaltennummer wobei bei 0 begonnen wird
- CMD = SORT
Sortiert eine MEMTAB
 - TABNAME = Eindeutiger Name der Tabelle die sortiert werden soll
 - P1 = Spaltenname 1
 - P2 = Sortierrichtung 1 (0 = Auf, 1 = Ab)
 - P3 = Spaltenname 2

- P4 = Sortierrichtung 2 (0 = Auf, 1 = Ab)
- P5 = Spaltenname 3
- P6 = Sortierrichtung 3 (0 = Auf, 1 = Ab)
- P7 = Spaltenname 4
- P8 = Sortierrichtung 4 (0 = Auf, 1 = Ab)
- P9 = Spaltenname 5
- P10 = Sortierrichtung 5 (0 = Auf, 1 = Ab)

- **CMD = GRUPPIEREN**
Erstellt eine neue, gruppierte MEMTAB
 - TABNAME = Eindeutiger Name der Tabelle, die gruppiert werden soll
 - P1 = Name der neu zu erstellenden Tabelle
 - P2 = Gruppenspalte (Name der Spalte, nach der gruppiert werden soll)
 - P3 = Gruppierungsart (Art, nach der numerischen Spalten gruppiert werden (0 = addieren, 1 = Maximum, 2 = Minimum))

- **CMD = GRUPPIERENERW**
Erweitertes Gruppieren
 - TABNAME = Eindeutiger Name der Tabelle die gruppiert werden soll
 - P1 = Name der neu zu erstellenden Tabelle
 - P2 = Gruppierungsart (Art nach der numerischen Spalten gruppiert werden (0 = addieren, 1 = Maximum, 2 = Minimum))
 - P3 = Bedingung (zusätzliche Selektion zur Prüfung, ob Zeilen berücksichtigt werden, z.B.: N0>0)
 - P4 = Gruppenspalte1 (Name der ersten Spalte, nach der gruppiert werden soll)
 - P5 = Gruppenspalte2 (Name der zweiten Spalte, nach der gruppiert werden soll)

- **CMD = GRUPPENWERT**
Rückgabe eines Gruppenwertes
 - TABNAME = Eindeutiger Name der Tabelle die gruppiert werden soll
 - P1 = Gruppenspalte (Name der Spalte, nach der gruppiert werden soll)
 - P2 = Gruppierungsart (Art nach der numerische Spalten gruppiert werden (0 = addieren, 1 = Maximum, 2 = Minimum, 3 = Durchschnitt))
 - P3 = Indexart (0 = Zeilennummer, 1 = Primärindex)
 - P4 = Rückgabebetyp (0 = gruppierter Wert, 1 = Zeilennummer (bei minimaler/maximaler Wert))
 - P5 = Indexart von (Zeilennummer/Primärindex ab dem Zeilen berücksichtigt werden)
 - P6 = Indexart bis (Zeilennummer/Primärindex bis zu dem Zeilen berücksichtigt werden)
 - P7 = Bedingung (zusätzliche Selektion zur Prüfung ob Zeilen berücksichtigt werden, z.B.: N0>0)

- **CMD = SCHREIBEN**
Schreibt den Inhalt der MEMTAB in eine Datei
 - TABNAME = Eindeutiger Name der Tabelle
 - P1 = Name des Files in das die Tabelle geschrieben werden soll
 - P2 = Fileformat (0 = CSV)

- **CMD = LESEN**
Liest den Inhalt der MEMTAB aus einer Datei
 - TABNAME = Eindeutiger Name der Tabelle
 - P1 = Name des Files aus dem die Tabelle gelesen werden soll
 - P2 = Fileformat (0 = CSV)

- **CMD = ZEILENANZAHL**
Liefert die Zeilenanzahl der MEMTAB
 - TABNAME = Eindeutiger Name der Tabelle

3.108 DUBLETENPRUEFUNG.EXEC

Führt eine Dublettenprüfung aus.

Parameter (Variante 1):

Es wird eine Dublettenprüfung für einen bestehenden Adressedatensatz durchgeführt.

Parameter	Beschreibung
ADRNR	Prüfung für die bestehende Adresse mit der angegebenen Adressnummer

Parameter (Variante 2):

Es wird eine Dublettenprüfung für durch die nachfolgenden Parameter definierte Adresse durchgeführt.

Parameter	Beschreibung
VORNAME	
NACHNAME	
NAMEZUSATZ	
STRASSE	
PLZ	
ORT	
TELEFONNUMMER	
MOBILNUMMER	
EMAILADRESSE	

Parameter (optional):

Parameter	Beschreibung
SCHWELLWERT	Distanzwert, ab dem eine Adresse als Dublette klassifiziert wird.
MAX_DUBLETTEN	Maximale Anzahl zurückgegebener Treffer.
PRUEFBEREICH	KOMPLETT (Standard), KUNDEN, LIEFERANTEN oder ERSTKONTAKTE

Rückgabe:

Das Prüfprotokoll wie im nachfolgenden Beispiel:

```
"DUBLETTENLISTE":  
{  
  "ADRESSE":  
  [  
    {  
      "ADNRN": "10001",  
      "DISTANZWERT": "100"  
    },  
    {  
      "ADNRN": "45678",  
      "DISTANZWERT": "67"  
    },  
  ],  
  "ANZAHL": "2"  
}
```

3.109 KASSENSTAMM.GET

Parameter	Beschreibung
KASNR	
VON_KASNR	
BIS_KASNR	

3.110 KASSENSTAMM.PUT

Parameter	Beschreibung
KASNR	

3.111 EINENTSTAMM.GET

Parameter	Beschreibung
EIENR	
VON_EIENR	
BIS_EIENR	

3.112 EINENTSTAMM.PUT

Parameter	Beschreibung
EIENR	

3.113 EINENTSTAMM.INSERT

Parameter	Beschreibung
EIENR	

3.114 EINENTSTAMM.DELETE

Parameter	Beschreibung
EIENR	

3.115 STEUERSCHLUESSEL.GET

Parameter	Beschreibung
STSNR	
VON_STSNR	
BIS_STSNR	

3.116 STEUERSCHLUESSEL.PUT

Parameter	Beschreibung
STSNR	

3.117 SACHKONTO.GET

Parameter	Beschreibung
FSANR	
VON_FSANR	
BIS_FSANR	

3.118 SACHKONTO.PUT

Parameter	Beschreibung
FSANR	

3.119 SACHKONTO.INSERT

Parameter	Beschreibung

3.120 SACHKONTO.DELETE

Parameter	Beschreibung
FSANR	

3.121 KONTOGRUPPE.GET

Parameter	Beschreibung
FGPNR	
VON_FGPNR	
BIS_FGPNR	

3.122 KONTOGRUPPE.PUT

Parameter	Beschreibung
FGPNR	

3.123 KONTOGRUPPE.INSERT

Parameter	Beschreibung

3.124 KONTOGRUPPE.DELETE

Parameter	Beschreibung
FGPNR	

3.125 PERSONAL.GET

Parameter	Beschreibung
PERNR	
VON_PERNR	
BIS_PERNR	

3.126 PERSONAL.PUT

Parameter	Beschreibung
PERNR	

3.127 PERSONAL.INSERT

Parameter	Beschreibung

3.128 PERSONAL.DELETE

Parameter	Beschreibung
PERNR	

4 Anhang

4.1 Beispiel für Inline-Positionsdaten bei BELEG.INSERT

```
{
  "WWSVC_FUNCTION":
  {
    "FUNCTIONNAME": "BELEG.INSERT",
    "POSDATEN":
    [
      {
        "PARAMETER":
        [
          {
            "PCONTENT": "Artikel19Prozent",
            "POSITION": "1",
            "PNAME": "POS_18_25"
          },
          {
            "PCONTENT": "1",
            "POSITION": "2",
            "PNAME": "POS_164_8"
          }
        ]
      },
      {
        "PARAMETER":
        [
          {
            "PCONTENT": "Artikel7Prozent",
            "POSITION": "1",
            "PNAME": "POS_18_25"
          },
          {
            "PCONTENT": "22",
            "POSITION": "2",
            "PNAME": "POS_164_8"
          }
        ]
      },
      {
        "PARAMETER":
        [
```

```
{
  {
    "PCONTENT": "K",
    "POSITION": "1",
    "PNAME" : "POS_17_1"
  },
  {
    "PCONTENT": "Testtext",
    "POSITION": "2",
    "PNAME": "POS_45_60"
  }
]
},
],
"PARAMETER":
[
  {
    "PCONTENT": "N",
    "POSITION": "1",
    "PNAME": "BEL_2_1"
  },
  {
    "PCONTENT": "10000",
    "POSITION": "2",
    "PNAME": "BEL_11_8"
  },
  {
    "PCONTENT": "10.02.2017",
    "POSITION": "3",
    "PNAME": "BEL_19_10"
  },
  {
    "PCONTENT": "09.02.2017",
    "POSITION": "4",
    "PNAME": "BEL_29_10"
  }
]
}
}
```

5 Übersicht der Änderungen

Version	Datum	Beschreibung
Rev. 1.0	11.02.2016	Referenz Servicefunktionen aktualisiert.
Rev. 1.1	09.03.2016	Referenz Servicefunktionen aktualisiert.
Rev. 1.2	24.03.2016	Referenz Servicefunktionen aktualisiert.
Rev. 1.3	13.04.2016	Referenz Servicefunktionen aktualisiert.
Rev. 1.4	18.04.2016	Referenz Servicefunktionen aktualisiert.
Rev. 1.5	11.05.2016	Referenz Servicefunktionen aktualisiert: GET_RELATION, PUT_RELATION, SUCHE_VOLLTEXT, FREISELEKT
Rev. 1.6	01.07.2016	Referenz Servicefunktionen aktualisiert: MIT_LANGTEXT, VERTRETER, WORKFLOW, DBK, DBP
Rev. 1.7	31.08.2016	Referenz Servicefunktionen aktualisiert: MIT_LAGERBESTAND, MIT_KATEGORIEN, MPKATALOG, MEDIEN, OHNE_LEERFELDER
Rev. 1.8	05.09.2016	Referenz Servicefunktionen aktualisiert: MEDIEN
Rev. 1.9	08.09.2016	Abschnitt über Selektionsausdrücke eingefügt
Rev. 1.10	14.09.2016	ARTIKGEL.GET erweitert für Artikelattribute
Rev. 1.11	28.09.2016	Einführung der Statistikfunktionen: STATARTLG.EXEC, STATARTUM.EXEC, STATLGDAT.EXEC, STATADR.EXEC, STATAAR.EXEC, STATPRJ.EXEC, STATLFA.EXEC, STATWGR.EXEC, STATVTWGR.EXEC
Rev. 1.12	25.10.2016	*.GET: Einführung der Parameter zum Holen genau eines

		Datensatzes (ARTNR, ADRNR, etc.)
Rev. 1.13	01.12.2016	*.GET: Freie Sortierung, BEDIENER.GET, FILE.GET, ARTIKEL.GET: Varianteninformationen
Rev. 1.14	21.12.2016	IMPORT.EXEC
Rev. 1.15	15.03.2017	Korrektur BELEG.INSERT, Anhang
Rev. 1.16	30.06.2017	MEMTAB.EXEC
Rev. 1.17	08.09.2017	FILE.PUT
Rev.1.18	01.12.2017	Neue Ressource: GESPRACH mit den Funktionen GET, PUT, INSERT und DELETE
Rev.1.19	20.12.2017	Neue Standardparameter: FREISELEKT_KEY, FREISELEKT_VON_INDEX, FREISELEKT_BIS_INDEX
Rev.1.20	27.12.2017	Referenz Servicefunktionen aktualisiert: WARENGRUPPE.GET, WARENGRUPPE.INSERT, WARENGRUPPE.PUT, WARENGRUPPE.DELETE, LAGER.GET, LAGER.INSERT, LAGER.PUT, LAGER.DELETE
Rev.1.21	29.12.2017	Referenz Servicefunktionen aktualisiert: MPKATEGORIE.GET, MPKATEGORIE.PUT, MPKATEGORIE.INSERT, MPKATEGORIE.DELETE
Rev.1.22	02.01.2018	Parameter GANZTAG in TERMIN.GET, TERMIN.PUT, TERMIN.INSERT. Neue Funktionen: EANCODE.GET, EANCODE.PUT, EANCODE.INSERT, EANCODE.DELETE
Rev.1.23	11.01.2018	Parameter IMPORTMODELLNR in IMPORT.EXEC.
Rev.1.24	30.01.2018	DUBLETENPRUEFUNG.EXEC
Rev. 1.25	06.03.2018	Neuer Parameter für ARTIKEL.GET: MIT_ATTRIBUT_LANGTEXT
Rev. 2.0	N.N.	Layout überarbeitet