



## WEBWARE 2.0

WEBWARE 2.0 (REV 11543) © SoftENGINE GmbH 2006-2015

## Dokumentation



# WW SVC

Beschreibung der WW Service Schnittstelle

Rel 2.8 22.12.2016

# INHALTSVERZEICHNIS

Begriffe .....	2
<b>WWSVC ein Überblick.....</b>	<b>4</b>
<b>WWSVC Erste Schritte .....</b>	<b>5</b>
Registrieren Kunden Installation im SoftENGINE AppStore .....	5
Registrierung durchführen .....	6
ServicePass bzw. Zugriff auf SEAS entfernen.....	7
Registrierung bearbeitet ? .....	7
Aktivieren der WEB-Service Funktion in Ihrer WWKIS .....	7
Meine Hersteller/Anwendungen aus SoftENGINE AppStore.....	9
Neue Service Anwendung konfigurieren .....	10
Konfiguration der WWSVC-Prozess-Knoten.....	12
Wie erhalte ich einen Überblick über die WWSVC-Prozesse.....	14
Sitzungs-Übersicht .....	14
Prozess-Übersicht.....	14
Funktions und Pool-Übersicht.....	15
Prozess-Knoten (WWSVC32.exe).....	16
Welche Service-Funktionen werden im System ausgeführt .....	17
Welche Service-Funktionen wurden im System ausgeführt.....	17
Das WWSVC Pool-System.....	18
Zuordnung von WWSVC-Objekten zu Pool's .....	18
<b>Testen der Schnittstelle mit iWWSVC.js.....</b>	<b>19</b>
<b>WEBWARE iWWSVC.js .....</b>	<b>25</b>
Auslieferung.....	25
Zielbeschreibung.....	25
Aufbau der iWWSVC-JS Schnittstelle .....	25
<b>iWWSVC-js   Test-Anwendung / Oberfläche .....</b>	<b>26</b>
<b>iWWSVC-js   Basis-Schnittstelle .....</b>	<b>27</b>
A) Anwendungs-Verwaltung .....	27
B) Verwaltung von Service-Pässen.....	28
C) Anforderung eines Session-Token .....	29
D) Ausführen von WWSVC Funktionen .....	29
E) Import Funktion.....	30
F) Beschreibung der erwarteten CallBack-Funktion .....	30
<b>WEBWARE SVC webServices .....</b>	<b>31</b>
Wie sieht ein Zugriff auf WWSVC aus .....	31
Folgende Aufgaben sind für einen Erstzugriff notwendig .....	31
Notwendige Aufgaben für einen registrierten Client .....	31
Zugriffsmodus Synchron/Asynchron .....	31
WWSVC der ServicePunkt.....	31
<b>WWSVC/WWSERVICE Funktionen Schnittstellenbeschreibung .....</b>	<b>32</b>
REGISTER.....	32

VALIDATE.....	32
DERGISTER.....	32
CONNECT.....	32
CLOSE.....	32
CURSORCLOSE.....	32
GETASYNCRESLUT.....	32
OPTIONS.....	33
WEBPROXY.....	33
Globale Rückgabe Information: COMRESULT.....	33
<b>WWSVC/WWSERVICE - Funktionen im Detail.....</b>	<b>34</b>
Registrierung eines Client /WWSVC/WWSERVICE/REGISTER.....	34
Mögliche Rückgabe-Codes .....	35
Registrierung abgelehnt .....	35
Registrierung OK, Freigabe von Administrator ausstehend.....	35
Registrierung OK .....	36
Validierung ServicePass: /WWSVC/WWSERVICE/VALIDATE .....	37
Mögliche Rückgabe-Codes .....	37
Validierung schlägt fehl .....	37
Validierung OK.....	38
De-Registrierung eines Client /WWSVC/WWSERVICE/DEREGISTER.....	39
Mögliche Rückgabe-Codes .....	39
DeRegistrierung schlägt fehl.....	40
DeRegistrierung OK .....	40
Neue Sitzung mit Benutzer-Liste eines Client /WWSVC/WWSERVICE/CONNECT .....	41
Mögliche Rückgabe-Codes .....	41
Erfolgreiche Anmeldung:.....	42
Ablehnung der Anmeldung: .....	42
Beende die Verbindung eines Client /WWSVC/WWSERVICE/CLOSE .....	44
Mögliche Rückgabe-Codes .....	44
CLOSE OK .....	44
WWSVC-CURSOR Beenden /WWSVC/WWSERVICE/CURSORCLOSE/ .....	45
Mögliche Rückgabe-Codes .....	45
CLOSE OK .....	45
Ergebnis asynchroner Service-Funktion abrufen /WWSVC/WWSERVICE/GETASYNCRESLUT .....	46
Mögliche Rückgabe-Codes .....	46
Bearbeitung ist abgeschlossen.....	48
Bearbeitung ist noch nicht abgeschlossen (HTTP-Code 202).....	49
Zugriff fehlgeschlagen (HTTP-Code 400).....	49
Verbindungrouting von Browser zu Fremdsystem /WWSVC/WWSERVICE/WEBPROXY.....	50
Mögliche Rückgabe-Codes .....	50
Beispiel Afterbuy Abruf.....	51
WWSVC/EXEC Funktion .....	52
HTTP / REST Funktionen.....	52
Hinweis bei Verwendung von Benutzer-Listen .....	52
Hinweis für die Bearbeitungsart (Synchron/Asynchron) .....	53
Hinweis für die Verwendung von WWSVC-Cursor .....	53
WWSVC/EXECURL Funktion.....	55
Beispiel der URL-Parameter für einen Funktionsaufruf: .....	55
Hinweis zur Verwendung und Angabe von Parametern:.....	55

Position/Parameter-Name im Aufruf .....	55
Angabe von Texten und dem URL-Trenner und Gleichheitszeichen .....	55
Aufruf /WWSVC/EXECURL .....	56
Aufruf /WWSVC/EXECURL .....	56
Antwort auf /WWSVC/EXECURL .....	56
WWSVC/EXECJSON Funktion .....	57
WWSVCEXEC JSON-Objekt für Funktionsaufruf .....	57
Mögliche Rückgabe-Codes .....	58
Cookies und Transport Sicherheit.....	59
WWSVC-REQID laufende Request-ID .....	59
WWSVC-TS Aktueller Zeitstempel .....	59
WWSVC-HASH Meldungs Hash für Validierung .....	59
WWSVC-APPID Name für Client-Anwendung .....	60
WWSVC-PASSID Service-Pass optional HTTP-Header.....	60
WWSVC-ACCEPT-RESULT-TYPE Form der Rückgabedaten.....	60
WWSVC-ACCEPT-RESULT-MAX-LINES Ergebnis einschränken.....	60
WWSVC-CURSOR Cursorfunktionen.....	61
WWSVC-DYNOBJ Verwendung Dynamische Objekte .....	61
WWSVC-SESSION-TOKEN .....	62
WWSVC-EXECUTE-MODE.....	63
WWSVC-CURSOR Beispiel Abruf .....	64
Funktionsaufruf mit WWSVC-Cursor-Anforderung .....	64
WWSVC-Cursor Ergebnis nach erstem Erfolgreichem Abruf .....	64
Folgeabruf eines WWSVC-Cursors .....	64
WWSVC-Cursor Beenden bei Lesen letztem Satz .....	65
WWSVC-Cursor Schließen vom Client aus .....	65
Fehler beim Abruf von WWSVC-CURSOR .....	65
WWSVC-DYNOBJ Dynamisches Objekt Beispiel Abruf.....	66
Antwort auf die Erstellung eines Dynamischen Objektes.....	66
Zugriff mit gültigem Dynamischen Objekt Token.....	66
Zugriff mit ungültigem Dynamischen Objekt Token.....	66
<b>Übersicht der Fehler/Status-Codes .....</b>	<b>67</b>
HTTP-Status-Codes .....	67
Status - OK.....	67
Status-Fehler .....	67
Server-Status-Fehler.....	68
Liste der Rückgabe-Codes .....	69
INFO 0 .. 9999 .....	69
WARNUNGEN 10000 .. 49999.....	69
FEHLER 50000 .. 99999 .....	69
<b>Aktivieren der WEBWARE Services im SystemCockpit .....</b>	<b>71</b>
Aktivieren von WWSVC.....	71
Die Bedeutung der Konfiguration Einträge .....	72
WW Service System Aktiv .....	72
WW Service über Internet erlaubt.....	72
WW Service Internet immer Admin Freigabe .....	72
WW Service über Intranet erlaubt.....	72
WW Service Intranet immer Admin Freigabe .....	73
Servicefunktion Wiederholzeit in Sekunden bei Restart.....	73
Setze diesen Mandant wenn keine Vorgabe, 0=Standard-Mandant .....	73
Starte WWSVC nur für diesen Mandanten, 0=Alle .....	73
Detail-Level COMRESULT-Struktur 0=Aus bis 9=Voll .....	73
WW Service CORS für Origin erlauben .....	74
WW Service CORS für alle Domains erlauben .....	74
WW Service ASYNCHRONE Zugriffe erlauben .....	74

WW Service ASYNCHRONE Ergebnisse Vorhaltezeit (Sekunden).....	74
Service-Zugang erlaubt Standard-Netzwerk Zugang .....	74
Service-Zugang erlaubt Zusatz-Netzwerk Zugang.....	74
Funktionsnamen aus HTTP-Befehlen ergänzen.....	75
Setze RESULT_MAX_LINES bei 0 auf (System-Wert).....	75
Verbindungsrouting von Browser zu Fremdsystem /WWSVC/WWSERVICE/WEBPROXY.....	76
CURSOR Verwendung erlaubt .....	77
CURSOR Beenden nach Sekunden Inaktivität .....	78
CURSOR: Zu einem Zeitpunkt nur 1x pro ServicePass .....	78
DYNOBJ Dynamische Objekte Verwendung erlaubt.....	78
DYNOBJ Beenden nach Sekunden Inaktivität .....	78
DYNOBJ Zu einem Zeitpunkt nur 1x pro ServicePass.....	78
<b>Verwaltung von WWSVC-Elementen .....</b>	<b>79</b>
Service Anwendungen (SecuredApp) .....	81
Folgende Parameter einer Service-Anwendung sind zu Bearbeiten .....	82
Verwalten von Benutzer-Listen .....	85
Verwalten von Benutzer-Listen im Bereich Administration.....	85
Service Pässe.....	86
Warten auf Freigabe.....	86
Alle Gültigen Service-Pässe / Alle aktiven Service-Pässe .....	88
Gesperrte/Abgelaufene Service-Pässe.....	88
<b>Referenz der Servicefunktionen .....</b>	<b>89</b>
<b>Übersicht Änderungen an diesem Dokument .....</b>	<b>92</b>

# WEBWARE WW SVC WEB Services

---

In diesem Dokument wird das HTTPs WEBSERVICE Interface der WEBWARE (WW SVC) definiert und beschrieben.

**Begriffe**

WW-SVC	WEBWARE Services (WEB Services der WEBWARE)
WW-KIS	WEBWARE Kunden Installation
WW-SEAS	WEBWARE SoftENGINE AppStore
HTTP	Protokoll für den Zugriff und Datenaustausch zwischen Client und WW-System
ServicePunkt	Der Zugangspunkt WWSVC erweitert die Domain <a href="https://myDomain.de/WWSVC">https://myDomain.de/WWSVC</a>
Hersteller	Erstellt Anwendungen die über WW-SVC Schnittstelle ausgeführt werden.  Der Hersteller wird mit einer eindeutigen 32-Byte-Hash-ID sowie Name definiert.
Anwendung	Eine Anwendung besteht aus einer definierten Funktionsgruppe und Rahmenparametern. Eine Anwendung wird mit einer eindeutigen 32-Byte-Hash-ID definiert.
SecuredApp	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 SecuredApp 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 SecuredApp 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 <a href="https://WWSVC/[PASS-ID]/Funktions-Name..">../WWSVC/[PASS-ID]/Funktions-Name..</a>
Parameter	Parameter können direkt beim Aufruf in der URL (URL-Encoded) übergeben werden. <a href="https://WWSVC/[PASS-ID]/MeineFunktion/10001/SORT=KNDNR">../WWSVC/[PASS-ID]/MeineFunktion/10001/SORT=KNDNR</a> Parameter können auch direkt im Body der WWSVC-Anfrage übergeben werden.
JSON	JavaScript Object Notation: Datenformat im Browser-Bereich
XML	eXtended Markup 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 Test:	Entwickler-Interface Test-Implementierung für WWSVC mit Javascript <a href="https://meine-webware.de/svc-js/svcjs.html">https://meine-webware.de/svc-js/svcjs.html</a>
CURSOR	Mit einem Cursor kann ein "Seitenweiser"-Sequentieller Zugriff auf einen großen Datenbestand erfolgen. Seitenweise bedeutet nach Vorgabe der Anzahl Ergebnissätze.



Dynamische Objekte      Bei Verwendung von Dynamisch erzeugten Objekten, wie zum Beispiel MEMTAB, kann man mit Hilfe von Dynamischen Objekt Token auf solche Dynamische erstellen Objekte zugreifen. Es wird dabei sichergestellt das die dynamischen Objekte beim Zugriff auch bei mehreren Prozessknoten verfügbar sind.

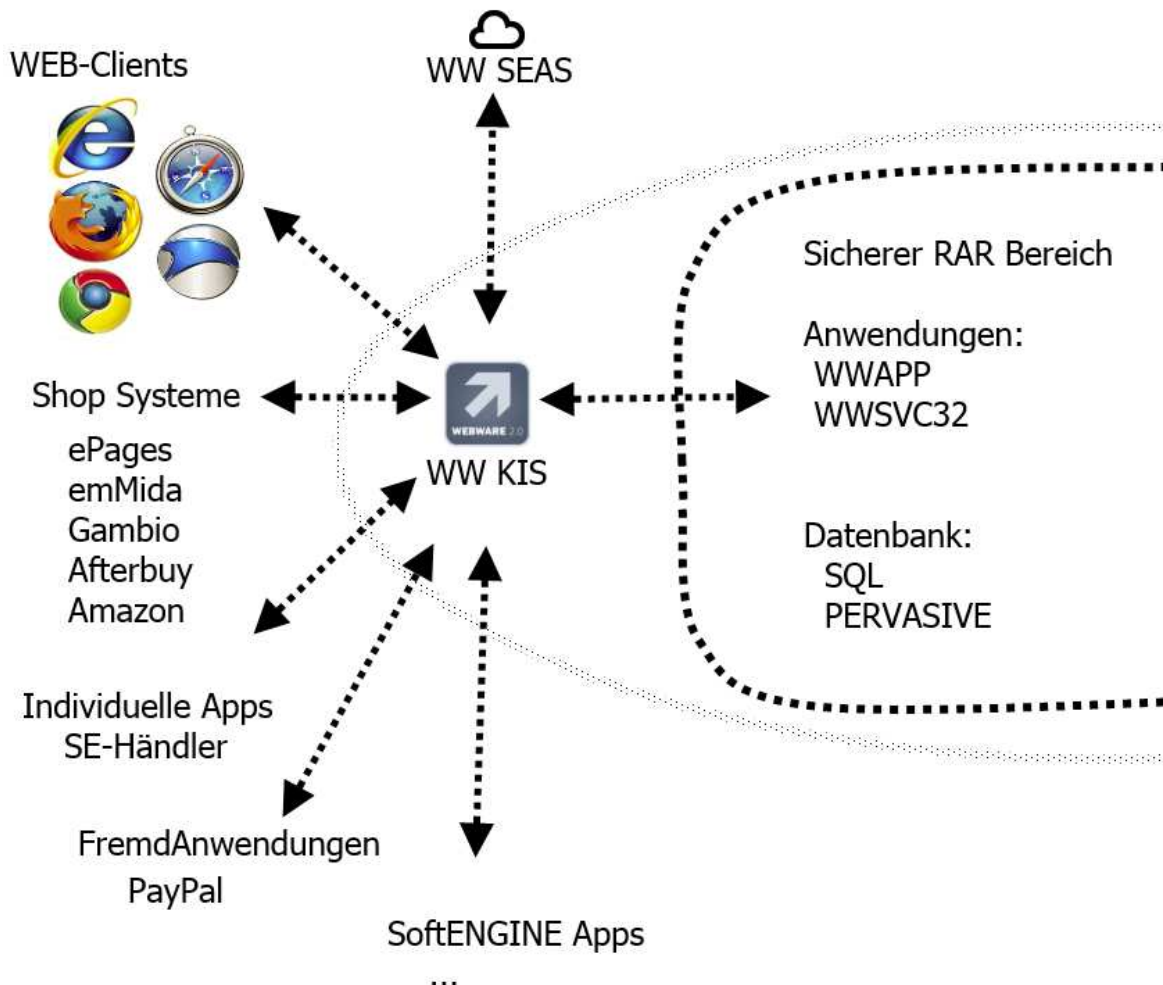
Zugriffsarten für WEBSERVICES:

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

ASYNCHRON      Service-Funktion wird in System eingelastet, das Ergebnis kann zu einem späteren Zeitpunkt mit Hilfe des erhaltenen Asynchron-Result-Handle abgerufen werden.

ASYNCHRON\_NO\_RESULT      Service-Funktion wird im System eingelastet. Das Ergebnis wird nicht benötigt und daher nach Abschluss der Service-Funktion nicht vorgehalten.

## WWSVC ein Überblick



Die **WEBWARE WEB Services WWSVC** erlauben den Zugriff auf Daten, Funktionen und Prozesse Ihrer WEBWARE Installation aus Anwendungen (**WWSVC-Apps**) von Dritt-Herstellern heraus.

Da die Unternehmensdaten / Funktionen / Prozesse sicherheitskritisch sind, wurde hier auf die Art und Form des Zugriffs unter Sicherheitsaspekten höchster Wert gelegt.

Die Verwaltung von **Hersteller/Anwendungen (WWSVC-Apps)** wird im **SoftENGINE AppStore SEAS** vom **SE-Händlern** durchgeführt und konfiguriert.

**WEBWARE Kunden Installationen WWKIS** welche WEB-Services nutzen wollen, registrieren sich direkt im **SEAS** und laden von dort ihre **Hersteller/Anwendungen** in ihre **WWKIS** herunter.

Der Zugriff auf WWSVC-Funktionen ist für Fremdanwendungen (Clients) nur über **WWSVC-Apps** möglich die zuvor vom WEBWARE System Administrator der **WWKIS** als "Meine **Service Anwendungen**" freigegeben wurden.

Jeder Client der auf eine freigegebene "**Service Anwendung**" zugreifen will, muss sich hierfür einmalig registrieren und erhält dadurch einen **Service-Pass** mit dem er dann auf die bereit gestellten Funktionen/Prozesse/Daten der "**Service Anwendung**" zugreifen kann.

Die Abarbeitung der Service-Funktionen wird dabei skaliert im RAR-Bereich über die WWSVC32-Anwendungen durchgeführt.

## WWSVC Erste Schritte

Im folgenden werden die Schritte erläutert die für die Inbetriebnahme der WEBWARE Web Services (WWSVC) in einer WWKIS (Kunden Installation) notwendig sind

- Registrieren Ihrer Kunden Installation im SoftENGINE AppStore
- Aktivieren der WEB-Service Funktion in Ihrer Kunden Installation
- Abruf der Hersteller/Anwendungen aus dem SoftENGINE AppStore
- Konfigurieren/Freigabe von "Meine Service Anwendungen"
- Konfigurieren der WWSVC-Prozess-Knoten
- Wie erhalte ich einen Überblick über die WWSVC-Prozesse
- Testen der Schnittstelle

## Registrieren Kunden Installation im SoftENGINE AppStore

Was? : Beschreiben System, und Registrieren bei SEAS

Kurz : Einstieg in System-Cockpit Installation, Vorgeben der Parameter für APP-Store und Registrieren

Um die WWSVC in Ihrer WEBWARE Kunden Installation (WWKIS) in Betrieb nehmen zu können, steigen Sie bitte in das WEBWARE System-Cockpit, im Bereich Installation ein. Sie finden die SEAS Konfiguration unter,

Installation > System Konfiguration > System Basis Konfiguration > SEAS SoftENGINE AppStore

The screenshot shows the 'SEAS SoftENGINE AppStore' configuration window. At the top, there's a header 'WEBWARE 2.0 für Meine Firma GmbH [01.2015-12.2015]'. Below it, a 'Meine Daten' tab is active. A button 'Ihr WEBWARE System registrieren' is visible. The main content area is divided into 'Selektion' (left) and 'Daten' (right). The 'Daten' section contains a table with system parameters.

Beschreibung	Systemwert
Name Ihres WW-System's im SE AppStore	Splash Bade-Moden WW 2.0
SEAS Service Pass Status	0
SEAS Service Pass	
SEAS SP_Datum	
SEAS SP_Uhrzeit	
SE APP-STORE ID (Anmeldeinformation)	Entwicklung@softengine.de
SE APP-STORE Passwort	*****
Optionaler ServicePass Benutzer	SEAS-100600
Optionales ServicePass Passwort	*****

Um Ihr System mit SEAS zu verbinden geben Sie Ihre eigenen Parameter vor:

Daten	
Beschreibung	Systemwert
Name Ihres WW-System's im SE AppStore	Splash Bade-Moden WW 2.0
SE APP-STORE ID (Anmeldeinformation)	Entwicklung@softengine.de
SE APP-STORE Passwort	*****

### ***Name Ihres WW-System's im SE AppStore***

Geben Sie hier den Namen an, unter dem Sie ihre WEBWARE Server Instanz im SEAS verwalten wollen.

### ***SE APP-STORE ID (Anmeldeinformation)***

Geben Sie den SoftENGINE-APP-Store-Benutzer an für den Sie diese WEBWARE Server Instanz registrieren wollen.

### ***SE APP-STORE Passwort***

Geben Sie hier das Passwort für den SoftENGINE-APP-Store-Benutzer an

## **Registrierung durchführen**

Mit Klick auf den Befehl "Ihr WEBWARE System Registrieren" können Sie ihre aktuelle WEBWARE Server Instanz beim SEAS registrieren.

Im ersten Schritt wird dabei eine Verbindung mit dem SEAS aufgebaut und ein Service-Pass für die folgenden Zugriffe auf SEAS erstellt.

Wurde ein ServicePass für den Zugriff erstellt, so wird der Service Pass Status auf 2 gesetzt sowie der Service-Pass ausgefüllt.

Die Angabe des Optionalen Service-Benutzer bzw. Service-Pass Passwortes ist nur nach Vorgabe von SoftENGINE notwendig.

Daten	
Beschreibung	Systemwert
SEAS Service Pass Status	2
SEAS Service Pass	1f70bdee3264b50e20fd2713b52f62
Optionaler ServicePass Benutzer	SEAS-100600
Optionales ServicePass Passwort	*****

### ServicePass bzw. Zugriff auf SEAS entfernen

Falls Sie den Zugang zum SEAS für diese WEBWARE Instanz löschen wollen, so können Sie dies mit dem Menü-Befehl



durchführen.

### Registrierung bearbeitet ?

Falls die Registrierung nicht sofort bearbeitet werden kann, so können Sie mit dem Menübefehl



prüfen ob die Registrierung bearbeitet wurde.

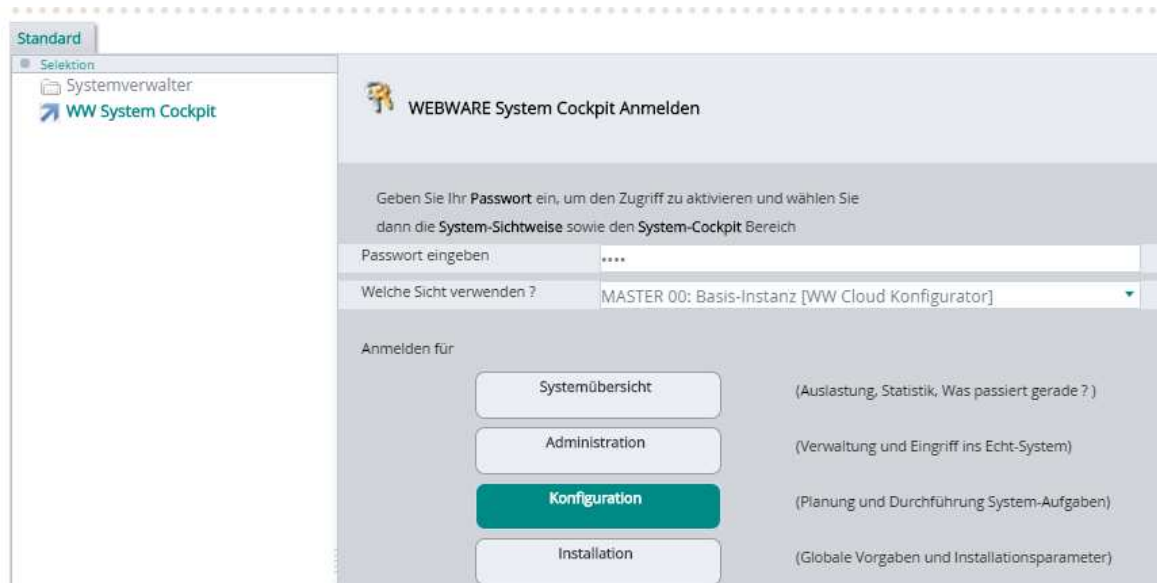
Nach Abschluss dieses Schrittes ist Ihre WW Kunden Installation (WWKIS) beim SoftENGINE AppStore (SEAS) registriert.

## Aktivieren der WEB-Service Funktion in Ihrer WWKIS

Was? : Aktivieren von WWSVC Funktion, sowie

Kurz : Einstieg in WW System-Cockpit Konfiguration, Sicherheits-Center und WWSVC aktivieren.

Um das WEBWARE Service System WWSVC für Ihr WEBWARE System zu aktivieren, melden Sie sich mit der Option Konfiguration an.



Sie finden die wichtigsten Parameter zur Aktivierung der WEBWARE Services (WW-SVC) im Bereich Sicherheits-Center. Sie können hier mit dem ersten Parameter die WW-SVC aktivieren. Mit den 4 weiteren Parametern können Sie die Zugriffs- und Freigabevorgaben für Inter- und Intranet vorgeben.



Weitere Parameter finden Sie unter "System Konfiguration" in dem Eintrag WWSVC WEBWARE Services Einträge.

Standard		Daten	
Selektion		Beschreibung	Systemwert
Systemverwalter		WW Service System Aktiv	1
WWSC Konfiguration 0-Basis-Instanz		WW Service über Internet erlaubt	1
System Übersicht		WW Service über Intranet erlaubt	1
Sicherheits Center		WW Service Internet immer Admin Freigabe	1
WW Cloud Instanzen		WW Service Intranet immer Admin Freigabe	1
System Prozesse		Servicefunktion Wiederholzeit in Sekunden bei Restart WW	30
System Laufzeitfunktionen anpa		Setze diesen Mandant wenn keine Vorgabe, 0=Standard	1
System Konfiguration		Starte WWSVC nur für diesen Mandanten, 0=Alle	1
WW-Programm Versionen		Detail-Level COMRESULT-Stuktur, 0=Aus bis 9=Voll	3
SUMO-Last-Simulation		WW Srvce CORS für Orignn erlauben	N
WW Externe Anbindungen		WW Srvce CORS für alle Domains erlauben	J
WW Zugangsschutz (SHIELD)		WW Service asynchrone Zugriffe erlauben	1
Benutzer Geräte mit AutoLogin		WW Service asynchrone Ergebnisse Vorhaltezeit (Sekunden)	3600
WW SVC WEBWARE Services			

Eine genaue Beschreibung der Parameter finden Sie im Kapitel ... weiter unten.

Am besten setzen Sie die Parameter wie oben angegeben.

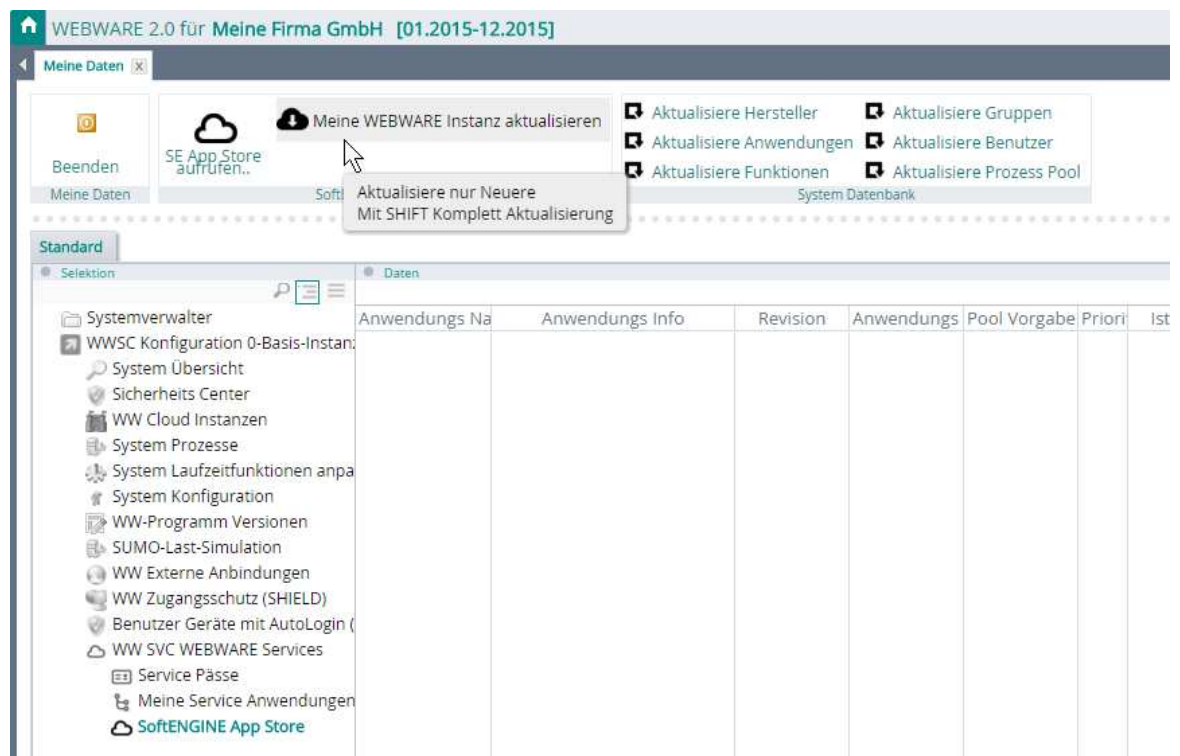
Nach Aktivierung der WWSVC Funktion muss der WW-Server Ihrer WWKIS neu gestartet werden.



## Meine Hersteller/Anwendungen aus SoftENGINE AppStore

Was? : Zugriff auf den SE-AppStore um meine Anwendungen ins WWKIS zu bringen.

Kurz : Einsteigen in System-Cockpit in Konfiguration, dann "Meine WWKIS aktualisieren" ausführen



Öffnen Sie das System-Cockpit und steigen Sie dort mit Konfiguration ein. Öffnen Sie ganz unten den Ast "WW SVC WEBWARE Services". Markieren Sie den Ast "SoftENGINE App Store".

Oben Im Menü erhalten Sie nun die Menü Gruppe: "SoftENGINE App Store" mit dem Befehl



Nach Klick auf diesen, wird eine Verbindung zum SoftENGINE App Store aufgebaut und die für Ihre Installation freigegebenen Anwendungen heruntergeladen.

Die Datenbank im RAR-Bereich wird nun aktualisiert und die Anwendungen/Funktionen/Regeln usw. heruntergeladen und bereit gestellt.

## Neue Service Anwendung konfigurieren

Was?: Festlegen wer, wie, wann auf was zugreifen kann.

Kurz: Einsteigen in System-Cockpit in Konfiguration, dann Hersteller/Anwendung für WWKIS freigeben

Öffnen Sie wie im vorigen Kapitel im Bereich System-Cockpit / Konfiguration den Ast "SoftENGINE App Store". Nach der "Aktualisierung" werden dort die für Ihre WWKIS freigegebenen Anwendungen angezeigt.

Nach Markierung einer Anwendung können Sie diese Anwendung im Datenbank Bereich bearbeiten ("Anwendung Bearbeiten"), also die Regeln/Vorgaben/Anforderungen der Funktionen für diese Anwendung ansehen und bearbeiten.

Anwendungs Na	Anwendungs Info	Revision	Anwendungs Pool	Vorgabe	Priori	Ist Erlaubt?	DB-Ist Erlaubt
Testanwendung	Testanwendung DevCamp 1.0	1					✓

Eine Benutzung dieser Hersteller/Anwendungen ist erst durch manuelle Freigabe/Konfiguration möglich. Um nun eine Hersteller/Anwendung für Ihre WWKIS zu aktivieren, markieren Sie diese und klicken Sie im Menü auf "Neue Service Anwendung konfigurieren".

Im folgenden Dialog ("WWSVC SecuredApp anlegen") werden nun die gewünschten Rahmen-Parameter für die Ausführung der Hersteller/Anwendung angegeben.

Eine Service Anwendung wird festgelegt indem der Ziel-Mandant sowie der Belegzeitraum definiert wird. Ziel der Service-Anwendung ist die Zugriffs-ID welche später vom Client benötigt um unter Angabe von Hersteller-ID und Anwendung-ID sowie der Zugriffs-ID einen Service-Pass anfordern zu können.

Der System-Administrator hat auch die Möglichkeit die Service-Anwendung auf bestimmte Benutzergruppen bei Registrierung und Ausführung/Betrieb einzugrenzen.

Durch aktivieren der Benutzerlisten Begrenzung beim Ausführen kann man auch eine Art Sitzungs-Token aktivieren, welcher den Zugriff erst nach Authentifizierung (Benutzer/Passwort) und für eine vorgebbare Zeit ermöglicht.



## WWSVC SecuredApp anlegen

Hersteller	SoftENGINE GmbH
Anwendung	Testanwendung DevCamp 1.0
Zugriffs-ID	2
Version	1
Mandant für Ausführung	2 : Beispieldaten
Begrenze Zeitraum	0 : Aktueller Zeitraum
Ausführungs Pool	STANDARD-POOL
Register Modus	1 : Erlaubt, Freigabe durch Admin
Benutzerliste bei Registrierung	<input checked="" type="checkbox"/>
Gruppe für Registrierung	1 : GRUPPE-1
Ausführungs Modus	1 : Ausführung erlaubt
Benutzerliste bei Ausführung	<input checked="" type="checkbox"/>
Gruppe für Ausführung	2 : GRUPPE-2
Max.Sitzungsdauer(Sec.)	86400
IntraNet nicht starten	<input type="checkbox"/>
InterNet nicht starten	<input type="checkbox"/>
Nur in NetSecureArea	<input type="checkbox"/>
Standardeintrag	<input checked="" type="checkbox"/>
Erlaubt von StartUhrzeit	<input type="text"/>
Erlaubt bis StartUhrzeit	<input type="text"/>
Setze Ablaufdatum	<input type="text"/>
Setze Ablauf in Tagen	<input type="text"/>

- Mandant für die Ausführung,

Eine Anwendung kann für einen Service-Pass immer nur auf einen Mandanten zugreifen, wollen Sie den Zugriff auf mehrere Mandanten müssen entsprechend mehrere Service-Anwendungen erstellt werden.

- Belegzeitraum für die Ausführung

Im Standard ist das Lesen in allen Belegzeiträumen möglich. Wollen Sie jedoch in einen anderen Belegzeitraum (Bspl. Vorjahr) schreiben, so muss dies hier in der Beschreibung der Service Anwendung angegeben werden.

- Ausführungs-Pool

Hier kann ein Programm-Pool vorgegeben werden, in dem die Service-Funktionen dieser Anwendung ausgeführt werden sollen.

- Modus für die Registrierung

Hier können Sie angeben ob bei der Service-Pass Registrierung dieser erst nach Freigabe durch den Admin, überhaupt nicht, oder sofort ohne Freigabe gültig wird.

- Optional: Registrierung nur für vorgegebene Benutzerliste

Angabe ob für die Anforderung eines Service-Passes ein Benutzer/Passwort mit übergeben werden soll, welches dann anhand der hinterlegten Benutzerliste geprüft wird. Dadurch wird auch der Service-Pass direkt mit dem Benutzernamen befüllt, welcher dann bei der Ausführung der Service-Funktionen verfügbar ist.

- Ausführungs-Modus, Erlaubt/Gesperrt

- Optional: Ausführung nur für vorgegebene Benutzerliste mit Sitzungsverwaltung  
Angabe ob die Ausführung nur erlaubt ist wenn der Benutzer zuvor ein Sitzungs-Token mit Benutzer/Passwort angefordert hat.

- Optional: Sitzungsdauer bei Sitzungsverwaltung  
Bei Anforderung von Sitzungstoken, kann hier die erlaubte Sitzungsdauer in Sekunden angegeben werden

- Zugang aus Internet/ Intranet erlaubt  
Hier kann der Zugang auf Internet / Intranet sowie bestimmte Netzbereiche begrenzt werden.

## Konfiguration der WWSVC-Prozess-Knoten

Was?: Konfiguration und Starten der Anwendungen welche die Service-Funktionen ausführen

Kurz: Einstieg in System-Cockpit Konfiguration dann Anwendung für WWSVC konfigurieren

Nachdem die WWSVC-Funktion aktiviert ist sowie Ihr WW-Server System neu gestartet ist, können Sie das System aktualisieren bzw. Service-Anwendungen konfigurieren.

Standard	
Selektion	Daten
Systemverwalter	
WWSC Konfiguration 0-Basis-Instanz	WW Service System Aktiv 1
System Übersicht	WW Service über Internet erlaubt 1
Sicherheits Center	WW Service über Intranet erlaubt 1
WW Cloud Instanzen	WW Service Internet immer Admin Freigabe 1
System Prozesse	WW Service Intranet immer Admin Freigabe 1
System Laufzeitfunktionen anpa	Servicefunktion Wiederholzeit in Sekunden bei Restz 30
System Konfiguration	Setze diesen Mandant wenn keine Vorgabe, 0=Stand 1
System Information	Starte WWSVC nur für diesen Mandanten, 0=Alle 1
System Basis Konfiguration	Detail-Level COMRESULT-Stuktur, 0=Aus bis 9=Voll 3
WWSVC WEBWARE Services	WW Srvce CORS für Originn erlauben N
STANDARD-POOL	WW Srvce CORS für alle Domains erlauben J
01 Standardmandant	WW Service asynchrone Zugriffe erlauben 1
02 Beispieldaten	WW Service asynchrone Ergebnisse Vorhaltezeit (Sel 20

Um aber Service-Funktionen auszuführen zu können muss nun noch die Prozess-Seite, also die Anwendungsseite konfiguriert werden. Hierzu steigen Sie im Bereich System-Cockpit Konfiguration in

*System-Cockpit > WWSC Konfiguration > System Konfiguration > WWSVC WEBWARE Services*

ein. Hier finden Sie auch die Vorgabeparameter für das WWSVC System. Unterhalb dieses Astes erhalten Sie eine Liste der Prozess-Pool's welche für die Verarbeitung der Service-Funktionen verwendet werden.

Das System stellt Ihnen zuerst einen Standard-Pool zur Verfügung. Sie können jedoch bei Bedarf weitere eigene Pool's erstellen welche dann auf die individuellen Bedürfnisse zugeschnitten werden können.

Neben der Maximal-Anzahl von gleichzeitig ausgeführten Prozess-Knoten, können hier auch Vorgabewerte für die maximale Wartezeit in den Pool-Warteschlangen für Start von Zusatz-Knoten, sowie maximale Leerlaufzeiten für das Herunterfahren von Zusatz-Knoten angegeben werden.

Um das ganze etwas einfacher zu machen verwenden wir den STANDARD-POOL und öffnen dessen Ast. Sie können nun für die Mandanten in Ihrem System und je Pool beliebig viele Anwendungen definieren die die Verarbeitung der Service-Anfragen übernehmen sollen.

Nachdem Sie einen Pool und den entsprechenden Mandanten ausgewählt haben erscheint in der Menüleiste der Knopf Erzeugen. Nach Klick auf diesen kommen Sie in die Maske um einen Prozessor-Knoten anzulegen.

Achtung: Änderung der Rahmenparameter von Prozessen haben direkt Auswirkung. Wird die Anzahl Basis-Knoten/Maximal-Knoten erhöht, so werden diese auch wenn nötig sofort gestartet. Bitte hier keine zu Hohen Werte eingeben (Also nicht 10000 Basis-Knoten starten lassen...)

Legen Sie hier nun fest wie der Prozessor-Knoten im Pool definiert werden soll.

- Prozessor Pool (Schlüssel, zu welchem Pool gehört der Knoten)
- Mandant (Schlüssel, für welchen Mandanten wird der Knoten betrieben)
- Beleg-Zeitraum (Schlüssel, in welchem Beleg-Zeitraum soll der Knoten laufen)
- Prozessor Type (Aktuell nur 32-Bit Version, später 64-Bit Version)
- Ausführungs-Server (Auf welchem RAR Server soll der Prozessor Knoten ausgeführt werden)
- Anzahl Basisknoten

Geben Sie hier die Anzahl Basisknoten an. Ein Basisknoten wird sofort gestartet und bleibt solange aktiv bis der Pool Beendet bzw. der Prozessor-Knoten Beendet wird. Basisknoten werden mit bei vor Zusatzknoten mit Service-Funktionen beschickt. Beispiel: 1= Es wird ein WWSVC32.exe gestartet

- Maximal Anzahl Knoten

Geben Sie hier die maximale Anzahl von Knoten an die für diesen Prozessor Knoten verwendet werden dürfen. Ist dieser Eintrag größer als die Anzahl Basisknoten, so werden je nach Auslastung, also Überschreitung der Wartezeit in der Warteschlange des Prozessor-Knotens weitere "Zusatz-Knoten" gestartet.

Gestartete Zusatz-Knoten werden nach Überschreiten einer Leerlaufzeit automatisch wieder Beendet.

Um eine bessere Auslastung der Basisknoten, sowie ein Leerlaufen der Zusatzknoten zu erreichen, kann eine Mindest-Wartezeit für den Pool angegeben werden, der für die Entscheidung Ausführung in Basis-Knoten oder Zusatz-Knoten verwendet wird.

## Wie erhalte ich einen Überblick über die WWSVC-Prozesse

Was?: Welche Prozesse laufen wie ist die Auslastung

Kurz: Einstieg in System-Cockpit Administration oder Konfiguration Bereich Sitzungen oder Prozesse sowie WW SVC WEBWARE Services

### Sitzungs-Übersicht

Steigen Sie in das System-Cockpit ein in den Bereich Administration ein. Hier finden Sie unter Sitzungen den Eintrag für WWSVC WEBWARE Services. Darunter werden die Pool's und Mandanten aufgelistet. Je nach Ast der gewählt wird, werden die entsprechenden Sitzungen angezeigt.

Standard		Daten											
Selektion		Status	Act.Speiche	Max.Speich	CPU%	Login a	Letzte	Erster Lo	Sitzur	IP	Port [VAr	Application	WWR-Server
Systemverwalter													
WWSVC Administration 0-Basis-Insta		RUN	23.83	38.96		11:23:3		11:23:37	11	19	17135	Ur WWSVC32	XL24D07BEC7APT
System Übersicht		RUN	35.41	36.29		17:37:2		17:37:22	23	19	11006	Ur WWSVC32	XL24D07BEC7APT
Sicherheits Center		RUN	35.23	35.54		17:37:2		17:37:22	24	19	11262	Ur WWSVC32	XL24D07BEC7APT
Sitzungen 22		RUN	35.25	35.54		17:37:2		17:37:22	25	19	11518	Ur WWSVC32	XL24D07BEC7APT
Anmeldebildschirm 0		RUN	35.19	35.42		17:37:2		17:37:22	26	19	11774	Ur WWSVC32	XL24D07BEC7APT
Aktiv 17		RUN	35.27	35.63		17:37:2		17:37:22	27	19	12030	Ur WWSVC32	XL24D07BEC7APT
Benutzer Sitzungen 1		RUN	35.25	35.46		17:37:2		17:37:22	28	19	12286	Ur WWSVC32	XL24D07BEC7APT
Öffentliche Sitzungen		RUN	35.35	35.54		17:37:2		17:37:22	29	19	12542	Ur WWSVC32	XL24D07BEC7APT
WW Interne Sitzungen 16		RUN	35.24	35.57		17:37:2		17:37:22	30	19	12798	Ur WWSVC32	XL24D07BEC7APT
WWSVC WEBWARE Servic		RUN	35.30	35.56		17:37:2		17:37:22	31	19	13054	Ur WWSVC32	XL24D07BEC7APT
STANDARD-POOL		RUN	35.25	35.43		17:37:2		17:37:22	32	19	13310	Ur WWSVC32	XL24D07BEC7APT
01 Standardmandant		RUN	35.49	35.66		17:37:2		17:37:22	33	19	13566	Ur WWSVC32	XL24D07BEC7APT
02 Beispieldaten													

### Prozess-Übersicht

Bei Einstieg in das System-Cockpit mit Administration bzw. Konfiguration können Sie im Bereich System-Prozesse ebenfalls die Anzeige der Pools und Mandanten jedoch in einer Prozess-Übersicht aufrufen.

Standard

Selektion

- Systemverwalter
  - WWSVC Administration 0-Basis-Insta
    - System Übersicht
    - Sicherheits Center
    - Sitzungen 22
      - WW Cloud Instanzen
      - Firmen
      - Protokoll-Dateien
      - WWR Anwendungs Server
      - System Prozesse
        - Haupt-System
        - WWR RAR-Server
        - Benutzer Sitzungen
        - WWSVC STANDARD-POOL
          - [ 11 ] WWSVC32-1 Mandant
          - [ 23 ] WWSVC32-2 Mandant
          - [ 24 ] WWSVC32-3 Mandant
          - [ 25 ] WWSVC32-4 Mandant
          - [ 26 ] WWSVC32-5 Mandant
          - [ 27 ] WWSVC32-6 Mandant
          - [ 28 ] WWSVC32-7 Mandant
          - [ 29 ] WWSVC32-8 Mandant
          - [ 30 ] WWSVC32-9 Mandant
          - [ 31 ] WWSVC32-10 Mandant
          - [ 32 ] WWSVC32-11 Mandant
          - [ 33 ] WWSVC32-12 Mandant
        - WWSVC SPEED-POOL
        - WWSVC SLOW-POOL
        - WWSVC SPEZIAL-POOL
        - WWSVC ASYNCHRON-POOL
        - WW Client Communicator

Details [ 11 ] WWSVC32-1 Mandant(2) Jahr(0)

Sitzungs-Zeiten

Performance

Interne Prozesse

IP-Setup

Application/Server

Versionsinformationen

WWSVC Konfiguration Prozessor

POOL Subsystem

Knoten-Type:

32-Bit Public WWSVC

Start-Modus:

Basis-Knoten

Basis Knoten für Knoten-Type:

12

Maximal Knoten für Knoten-Type:

15

Start-Nummer für Netz-Knoten:

1

Mandant für Knoten:

2

Belegzeitraum für Knoten:

0

Auslastung aktueller Prozess Knoten

Anzahl In Warteschlange:

1

Aktuelle Wartezeit:

0 Millisekunden

Aktuelle Verarbeitungszeit:

3 Millisekunden

Wartezeit Mandant/Belegzeitraum:

3 Millisekunden

Aktuelle Zustand:

RUNNING/PROCESSING

STANDARD-POOL Aktueller Zustand

Maximal Startbare Prozess-Knoten:

25

Anzahl verwendbare Basis-Knoten:

12

Anzahl gestartete Prozess-Knoten:

12

Anzahl startbare Zusatz-Knoten:

3

Anzahl Servicefunktionen im Pool:

1

Letzte Wartezeit beim einfügen:

3 Millisekunden

Starte Knoten wenn Wartezeit >

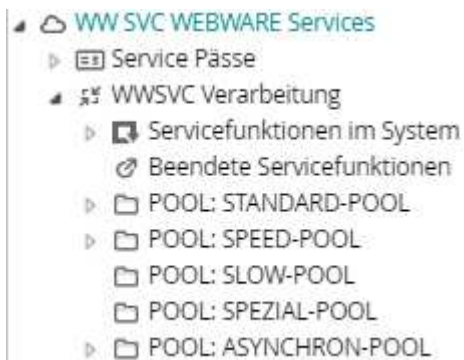
2000 Millisekunden

Beende Knoten wenn Leerlaufzeit >

60000 Millisekunden



## Funktions und Pool-Übersicht

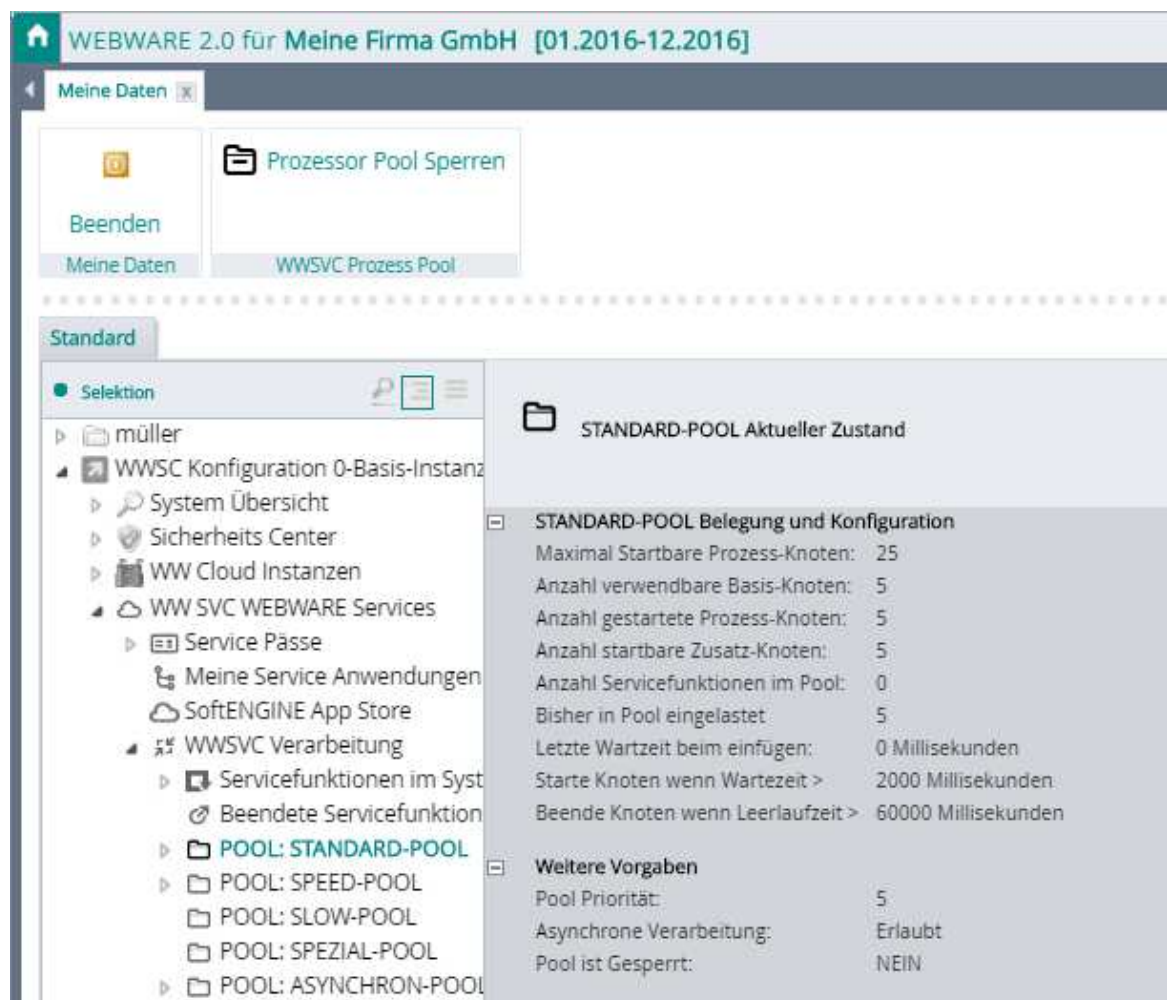


Im System-Cockpit in den Sichten Konfiguration und auch Administration gibt es im Bereich "WW SVC WEBWARE Services" den Ast WWSVC Verarbeitung.

Dort finden Sie alle aktuellen Laufzeitwerte für Service-Funktionen und auch Service-Pool und Service-Prozessoren.

Die Daten werden dabei in Echtzeit aktualisiert so dass Sie jederzeit den vollen Überblick über die Auslastung Ihres Systems haben.

Die aktuelle Auslastung sowie die Funktion zum Sperren und Freigeben von einzelnen Pool's finden Sie hier direkt bei Markierung eines Pool's.



## Prozess-Knoten (WWSVC32.exe)

Unterhalb der Pool's werden die aktuell laufenden Prozess-Knoten also die WWSVCxx.exe die je für einen Mandant und Belegzeitraum laufen aufgelistet. Markieren Sie im Baum einen solchen Ast, so erhalten Sie für diesen Prozess-Knoten die aktuelle Echtzeitübersicht der Belegung und Konfiguration angezeigt.

**WEBWARE 2.0 für Meine Firma GmbH [01.2016-12.2016]**

---

**Meine Daten**

Beenden  
Meine Daten

Prozess Abbrechen  
Service Knoten

---

**Standard**

**Selektion**

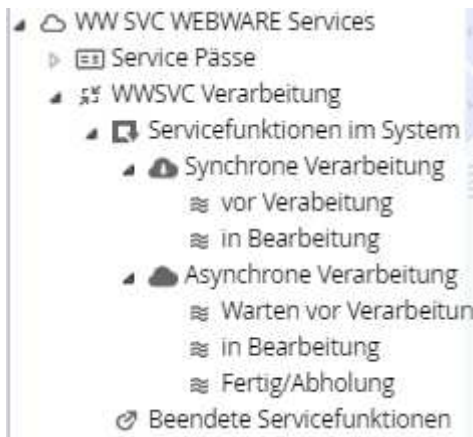
- müller
- ▾ WWSC Konfiguration 0-Basis-Instanz
  - System Übersicht
  - Sicherheits Center
  - WW Cloud Instanzen
  - System Prozesse
  - System Laufzeitfunktionen anpassen
  - System Konfiguration
  - WW-Programm Versionen
  - SUMO-Last-Simulation
  - WW Externe Anbindungen
  - WW Zugangsschutz (SHIELD)
  - Benutzer Geräte mit AutoLogin (P)
  - ▾ WW SVC WEBWARE Services
    - Service Pässe
      - Meine Service Anwendungen
      - SoftENGINE App Store
    - ▾ WWSVC Verarbeitung
      - Servicefunktionen im System
        - Beendete Servicefunktion
      - ▾ POOL: STANDARD-POOL
        - [ 15 ] WWSVC32-1 Mandant(2) Jahr(0)
        - [ 17 ] WWSVC32-3 Mandant(2) Jahr(0)
        - [ 18 ] WWSVC32-4 Mandant(2) Jahr(0)
        - [ 19 ] WWSVC32-5 Mandant(2) Jahr(0)
        - [ 20 ] WWSVC32-2 Mandant(2) Jahr(0)
      - POOL: SPEED-POOL
      - POOL: SLOW-POOL
      - POOL: SPEZIAL-POOL
      - POOL: ASYNCHRON-POOL

**Details [ 15 ] WWSVC32-1 Mandant(2) Jahr(0)**

<b>Sitzungs-Zeiten</b>	
<b>Performance</b>	
<b>Interne Prozesse</b>	
<b>IP-Setup</b>	
<b>Application/Server</b>	
<b>Versionsinformationen</b>	
<b>WWSVC Konfiguration Prozessor</b>	
POOL Subsystem	STANDARD-POOL
Knoten-Type:	32-Bit Public WWSVC
Start-Modus:	Basis-Knoten
Basis Knoten für Knoten-Type:	5
Maximal Knoten für Knoten-Type:	10
Start-Nummer für Netz-Knoten:	1
Mandant für Knoten:	2
Belegzeitraum für Knoten:	0
<b>Auslastung aktueller Prozess Knoten</b>	
Anzahl in Warteschlange:	0
Bisher eingelastet:	1
Aktuelle Wartezeit:	0 Millisekunden
Aktuelle Verarbeitungszeit:	0 Millisekunden
Wartezeit Mandant/Belegzeitraum:	0 Millisekunden
Aktuelle Zustand:	LEERLAUF seit 00:14:41
<b>STANDARD-POOL Aktueller Zustand</b>	
Maximal Startbare Prozess-Knoten:	25
Anzahl verwendbare Basis-Knoten:	5
Anzahl gestartete Prozess-Knoten:	5
Anzahl startbare Zusatz-Knoten:	5
Anzahl Servicefunktionen im Pool:	0
Bisher in Pool eingelastet:	5
Letzte Wartezeit beim einfügen:	0 Millisekunden
Starke Knoten wenn Wartezeit >	2000 Millisekunden
Beende Knoten wenn Leerlaufzeit >	60000 Millisekunden

Falls Sie im Bereich System-Konfiguration sind, haben Sie im Menü die Möglichkeit den markierten Prozess-Knoten zu Beenden. Dabei ist zu Beachten das dieser von der WWSVC-Verwaltung bei Bedarf sofort wieder mit einer neuen Sitzungs-Nummer gestartet wird.

## Welche Service-Funktionen werden im System ausgeführt



Sie finden in der Administration und Konfiguration des System-Cockpit unterhalb des Astes WW SVC WEBWARE Services den Bereich WWSVC-Verarbeitung.

Im Unterast Servicefunktionen im System können Sie die aktuellen sowie die Beendeten Service-Funktionen anschauen.

Dabei wird bei den aktuellen Service-Funktionen in Synchrone und Asynchrone Service-Funktionen unterschieden.

Eine Liste der letzten 128 ausgeführten Service-Funktionen finden Sie unterhalb von Beendete Servicefunktionen.

Die Service-Funktionen werden dabei in "Vor Verarbeitung", also noch nicht eingeplant, und "In Bearbeitung", also an einen Service-Prozessor zur Ausführung übergeben unterschieden.

Standard

Selektion

- müller
  - WWSC Administration 0-Basis-Instanz
    - System Übersicht
    - Sicherheits Center
    - WW SVC WEBWARE Services
      - Service Pässe
      - WWSVC Verarbeitung
        - Servicefunktionen im Sys
          - Synchrone Verarbeitung
            - vor Verarbeitung
            - In Bearbeitung
          - Asynchrone Verarbeitung
            - Warten vor Verarb
            - In Bearbeitung
            - Fertig/Abholung

Daten

Laufzi	Funktionsname	Prozess-Pool	Server Zustand	Prozess Zustand	ServicePass-ID	Anwendungs-ID	Hersteller-ID
5	ARTIKEL.GET	STANDARD-POOL	Gesendet in entfernten Knoten	Warte in entfernter Warteschlange	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85	
6	ARTIKEL.GET	STANDARD-POOL	Gesendet in entfernten Knoten	Warte in entfernter Warteschlange	c1d9136edf6424c58e309af4f42daci	da172c74e79c888	
7	ARTIKEL.GET	STANDARD-POOL	Gesendet in entfernten Knoten	Warte in entfernter Warteschlange	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85	

Im Bereich Asynchrone Bearbeitung gibt es noch zusätzlich den Ast, Fertig/Abholung unter dem die Asynchrone Service-Funktionen aufgelistet werden welche fertig bearbeitet sind, jedoch noch nicht vom Client abgerufen wurden.

Laufzi	Funktionsname	Prozess-Pool	Server Zustand	Prozess Zustand	Si
15	ARTIKEL.GET	ASYNCHRON-POC	Gesendet in entfernten Knoten	Bearbeitet, Ergebnis steht bereit	3b
17	ARTIKEL.GET	ASYNCHRON-POC	Gesendet in entfernten Knoten	Bearbeitet, Ergebnis steht bereit	c1
19	ARTIKEL.GET	ASYNCHRON-POC	Gesendet in entfernten Knoten	Bearbeitet, Ergebnis steht bereit	3b

## Welche Service-Funktionen wurden im System ausgeführt

Die letzten 128 ausgeführten Service-Funktionen werden im Bereich "Beendete Servicefunktionen" angezeigt.

Standard													
Selektion		Daten											
müller		- Laufzi	Funktionsname	Prozess-Pool	HTTP S/Start Datum	Start Uhrzeit	Ende Uhrzeit	ServicePass-ID	Anwendungs-ID	Hersteller-ID	Cursor-ID	Modus	Async-info
WWSC Administration 0-Basis-Instanz		11	ARTIKEL.GET	ASYNCHRON-POC200	2016.09.13	18:03:26-776	18:03:35-647	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	RESULT READY TAKEN
System Übersicht		13	ARTIKEL.GET	ASYNCHRON-POC200	2016.09.13	18:03:27-500	18:03:35-487	c1d9136edf6424c58e309af4f42daci	da172c74e79c888			ASY	RESULT READY TAKEN
Sicherheits Center		9	ARTIKEL.GET	ASYNCHRON-POC200	2016.09.13	18:03:25-728	18:03:32-093	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	RESULT READY TAKEN
Sitzungen 19		11	ARTIKEL.GET	ASYNCHRON-POC200	2016.09.13	18:03:26-776	18:03:35-647	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	
WW Cloud Instanzen		13	ARTIKEL.GET	ASYNCHRON-POC200	2016.09.13	18:03:27-500	18:03:35-487	c1d9136edf6424c58e309af4f42daci	da172c74e79c888			ASY	
Firmen		9	ARTIKEL.GET	ASYNCHRON-POC200	2016.09.13	18:03:25-728	18:03:32-093	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	
WW SVC WEBWARE Services		12	ARTIKEL.GET	ASYNCHRON-POC202	2016.09.13	18:03:25-728	18:03:32-093	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	
Service Pässe		10	ARTIKEL.GET	ASYNCHRON-POC202	2016.09.13	18:03:26-776	18:03:26-777	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	
WWSVC Verarbeitung		8	ARTIKEL.GET	ASYNCHRON-POC202	2016.09.13	18:03:25-728	18:03:25-729	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	
Servicefunktionen im System		7	ARTIKEL.GET	STANDARD-POOL 200	2016.09.13	17:57:57-394	17:58:04-205	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			SYN	
Synchrone Verarbeitung		6	ARTIKEL.GET	STANDARD-POOL 200	2016.09.13	17:57:56-409	17:58:03-209	c1d9136edf6424c58e309af4f42daci	da172c74e79c888			SYN	
vor Verarbeitung		5	ARTIKEL.GET	STANDARD-POOL 200	2016.09.13	17:57:55-990	17:58:02-784	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			SYN	
In Bearbeitung		3	ARTIKEL.GET	ASYNCHRON-POC200	2016.09.13	17:18:25-526	17:18:32-328	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	RESULT READY TAKEN
Asynchrone Verarbeitung		4	ARTIKEL.GET	STANDARD-POOL 200	2016.09.13	17:18:27-006	17:18:38-032	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			SYN	
Warten vor Verarbeitung		1	ARTIKEL.GET	STANDARD-POOL 200	2016.09.13	17:18:25-526	17:18:32-328	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	
In Bearbeitung		2	ARTIKEL.GET	STANDARD-POOL 200	2016.09.13	17:18:24-862	17:18:31-699	c1d9136edf6424c58e309af4f42daci	da172c74e79c888			SYN	
Fertig/Abholung		1	ARTIKEL.GET	ASYNCHRON-POC202	2016.09.13	17:18:25-526	17:18:25-527	3b2ee14877f3cdf	04abf1c38b8522853f69160a5b0b85			ASY	
Beendete Servicefunktion													

Falls Sie alle bearbeitete Service-Funktionen als EXCEL-Datei (.CSV) mit allen Informationen anschauen wollen, so gibt es im Pfad

bin\wvs\performance\[lfd-ServerStart-Nr]\WWSVCPERF-[Datum]-[Instanz-nr].CSV

für jeden Tag und WEBWARE-Instanz eine eigene CSV-Datei.

## Das WWSVC Pool-System

Die WEBWARE Services verwenden zur Gruppierung von Prozess-Knoten (WWSVCxx.exe) sogenannte Pools. Im System gibt es vordefinierte Pool's welche bei der Erstkonfiguration des Systems per Pool-Definition die Anzahl von Prozess-Knoten je Mandant und Belegzeitraum zugewiesen werden müssen.

Folgende Pool's sind in der Bassi-Installation vorgesehen:

STANDARD-POOL                      Standard-Ausführung mit normaler Priorität

ASYNCHRON-POOL                  Ausführung von Asynchronen Anfragen mit geringerer Priorität

Sie können hier selbst Pool's anlegen und damit die Reaktion auf Service-Funktionen Anfragen steuern.

## Zuordnung von WWSVC-Objekten zu Pool's

Sie können innerhalb der WWSVC bei unterschiedlichen Objekten einen Wunsch-Pool mit vorgeben, welcher bei der Ausführung der Service-Funktionen dann berücksichtigt wird.

Folgende Objekte erlauben es einen Ausführungs-Pool zu hinterlegen. Die Nennung erfolgt in hierarchischen Reihenfolge wie das System die Pool's verwendet.

Mögliche Pool-Vorgabe im

- Service-Pass
- Benutzer Satz
- Benutzer-Gruppen-Satz
- Funktions-Satz
- Secured-App
- Anwendung
- Hersteller



## Testen der Schnittstelle mit iWWSVC.js

Was?: Mit Hilfe der iWWSVC.js Test-Suite WWSVC Funktionen testen

Kurz: Aufruf [https://\[mein-Server-Domain.xx\]/svc-js/svc.js.html](https://[mein-Server-Domain.xx]/svc-js/svc.js.html)

Mit dem Interface WWSVC Javascript iWWSVC.js können Sie direkt aus einem Browser heraus die Verbindung zu einem Server sowie die Verwendung von Service-Pässen und Ausführung von Funktionen testen. Die iWWSVC.JS Anwendung ist so ausgelegt das sie in allen modernen Browsern funktioniert.

Um das Test-Programm zu Starten rufen Sie die URL Ihres WWKIS auf, sowie den Pfad

[https://\[mein-Server-Domain.xx\]/svc-js/svc.js.html](https://[mein-Server-Domain.xx]/svc-js/svc.js.html)

in einem Browser auf.

WWSVC Test Anwendung 1.03 / 21.02.2016

Active | Hersteller | Anwendung | Options +

Anwendungen

ServicePass

Active | Function | Exec | ExecExtra | Exec | ExecExtra | Options | +

Funktionen

Beim ersten Aufruf ist die Anzeige noch leer. Nach Anlage von Anwendungen, Service-Pässen bzw. Funktionen werden diese angezeigt und lokal in Ihrem Browser für die weitere Verwendung gespeichert.

Anlage einer "Service Anwendung durch Klick auf das PLUS-Symbol im Bereich Anwendungen:

WWSVC Test Anwendung 1.03 / 21.02.2016

WWSVC Anwendung konfigurieren X

Server-URL

Server-Port

Hersteller ID

Anwendungs ID

Secure-ID

Anwendungs Revision

Anwendung speichern

Geben Sie hier nun den Server-Zugang (URL/Port) sowie die Schlüssel-Daten der Service-Anwendung, also Hersteller-ID, Anwendungs-ID sowie Secure-ID (Zugriffs-ID). Optional kann noch die Revision. Im nächsten Schritt wird die Anwendung bei Ihrem WW-Server registriert. Dazu Klicken Sie auf den "Register"-Knopf in der Anwendungs-Zeile. Sie erhalten die Anzeige für den Service-Pass welchen Sie nun registrieren können. Optional wäre hier auch die Möglichkeit ein Benutzer und Passwort für die Registrierung zu übergeben.

Nach Klick auf Service-Pass registrieren, erhalten Sie die Anzeige des Ergebnisses. Dies ist für alle Aktionen ähnlich. Im oberen Bereich erhalten Sie eine Info über die Funktion und die folgenden Informationen:

**ServicePass Register**

**Funktion Start**  
Funktion wurde erfolgreich gestartet

**Funktions Ergebnis**  
Aufruf-Code: R  
HTTPCode: 202  
ResultCode: 1  
Fehler/Hinweistext: REGISTRATION OK, WAIT FOR ADMIN RELEASE

**Objekt**

```
{
  "json": {
    "COMRESULT": {
      "STATUS": 202,
      "CODE": "202 Accepted",
      "INFO": "REGISTER OK, WAIT FOR ADMIN RELEASE",
      "ERRORCODE": 10000,
      "ERRORLINK": "DOCHWSVC/INFO.HTML/#10000",
      "ERRORINFO": "REGISTER OK WAIT FOR ADMIN RELEASE"
    },
    "SERVICEPASS": {
      "PASSID": "db202809571ec48e069ff5b3c68c651c",
      "APPID": "27e3d07652461f934d90ed8569aacef4",
      "PDATE": 20160209,
      "PTIME": 18080188
    }
  }
}
```

Aufgerufene Funktion (ServicePass Register)

War der Start erfolgreich, oder wurde schon in iWWSVC.JS abgebrochen?

Ergebnis vom WW-Server in der Übersicht

(Hier Meldung WAIT FOR ADMIN RELEASE!!)

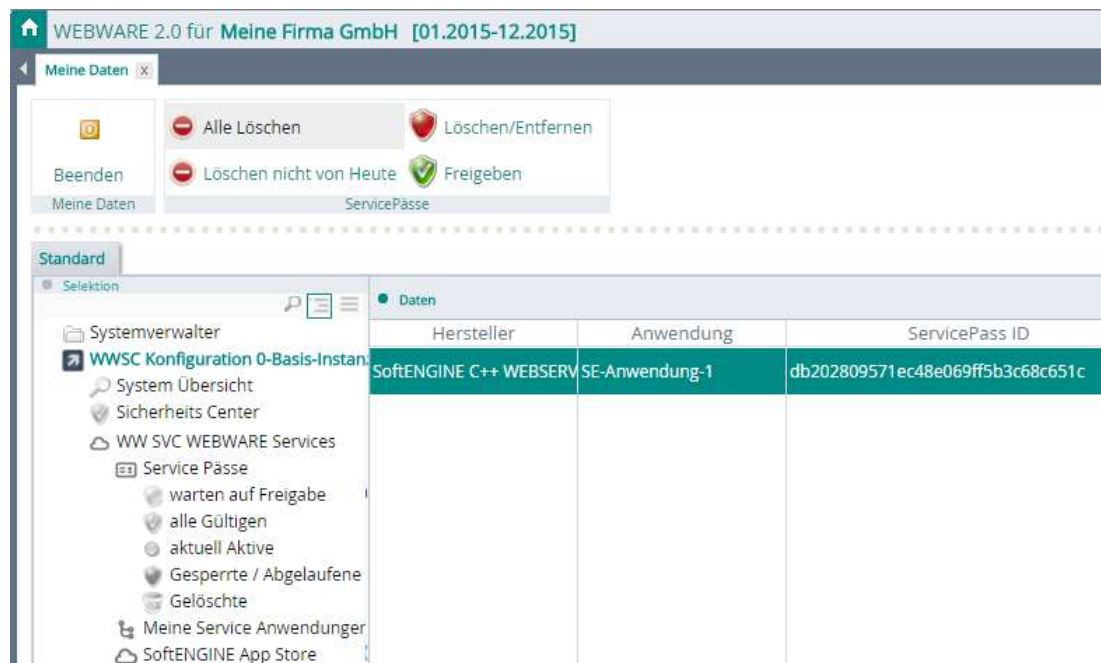
Im unteren Bereich erhalten Sie dann die Ausgabe des Body-Bereiches aus der Antwort. Falls es ein JSON-Objekt ist, wird dieses direkt aufbereitet angezeigt.

Ist die Antwort kein JSON-Objekt, wird hier das Objekt als eingebettetes Frame angezeigt.

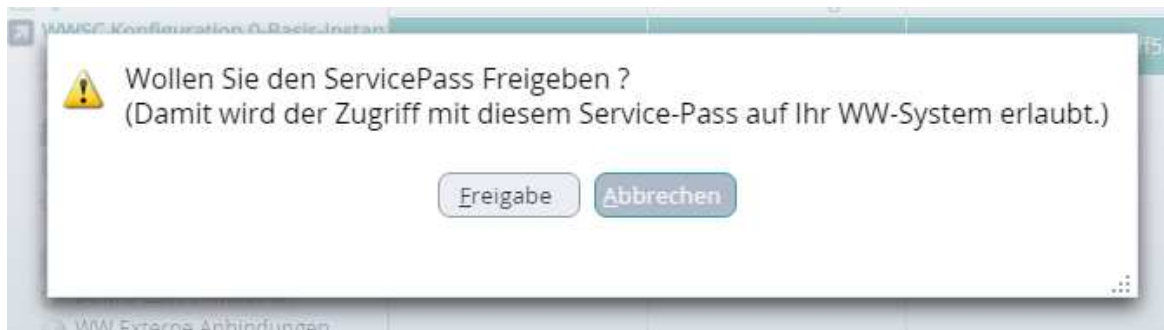
Der Service-Pass wurde erstellt, jedoch mit Status-Code 202 zurückgegeben. Dies bedeutet der Administrator muss den Service-Pass erst noch freigeben.

Wechseln Sie hierzu in das System-Cockpit in den Bereich Konfiguration.

*System-Cockpit: Konfiguration > WW-SVC-WEBWARE-Services > Service Pässe*



Hier erhalten Sie die Auflistung aller Service-Pässe. Im Ast "Warten auf Freigabe" finden Sie den Service-Pass welcher vom Administrator noch manuell freigegeben werden soll. Durch Klick auf Freigeben und Bestätigung der Abfrage wird der Service-Pass als Gültig markiert.



Wechseln Sie nun wieder zurück in die Test-Anwendung und Führen Sie im Bereich ServicePass den Befehl Validate aus:



ServicePass Validieren

Hersteller ID

53f69160a5b0b89136ba1c6390c1e5d1

Anwendungs ID

04abf1c38b8522869f857dcffa3c5500

Secure-ID

3

Anwendungs Revision

0

ServicePass

db202809571ec48e069ff5b3c68c651c

ServicePass Validieren

ServicePass Validate

Funktion Start

Funktion wurde erfolgreich gestartet

Funktions Ergebnis

Aufruf-Code: V

HTTPCode: 200

ResultCode: 2

Objekt

```
{
  "json": {
    "COMRESULT": {
      "STATUS": 200,
      "CODE": "200 OK",
      "INFO": "SERVICEPASS OK",
      "ERRORCODE": 200,
      "ERRORLINK": "DOCHWSVC/INFO.HTML/#00200",
      "ERRORINFO": ""
    }
  }
}
```

Nach Ausführung von ServicePass Validieren erhalten Sie das Ergebnis HTTP-Code 200 OK Für Service-Pass OK. Nun können Sie mit dem Service-Pass auf die Funktionen zugreifen.

Nachdem der Service-Pass erfolgreich eingetragen wurde, können Sie die Funktionsliste für diese Anwendung abrufen. Klicken Sie hierzu auf den Eintrag Funktionen Anfordern.

WWSVC Test Anwendung 1.03 / 21.02.2016

Active	Hersteller	Anwendung			Options
<input checked="" type="checkbox"/>	da172c74e79c8886bc5aa87979e953cg	58e309af4f42dac98ece7b958fee0e0	Edit	Delete	Abrufen

Anwendungen

Service-Pass	Session-Token	Pass	Pass	Session	Options
2eea5c6acda6aaa68135ddb130d53d18		Validate	DeRegister	Anfordern	Abrufen

ServicePass

Active	Function	Exec	ExecExtra	Exec	ExecExtra	OPTIONS
--------	----------	------	-----------	------	-----------	---------

Funktionen

Es werden dann die Funktionen für diese Anwendung heruntergeladen und in der Funktionsliste mit Parametern eingetragen.

Funktionsliste abrufen

Funktion Start

Funktion wurde erfolgreich gestartet

Funktions Ergebnis

Aufruf-Code: J

HTTPCode: 200

ResultCode: 1

Es wurden 3 Funktionen geladen

WWSVC Test Anwendung 1.03 / 21.02.2016

Active	Hersteller	Anwendung	Options		
<input checked="" type="radio"/>	da172c74e79c8886bc5aa87979e953cg	58e309af4f42dac98ece7b958fee0e0l	Edit	Delete	Abrufen

Anwendungen

---

Service-Pass	Session-Token	Pass	Pass	Session	Options
2eea5c6acda6aaa68135ddb130d53d18					Validate DeRegister Anfordern Abrufen

ServicePass

---

Active	Function			Exec	ExecExtra	Exec	ExecExtra	Options			
<input type="radio"/>	ARTIKEL[1]	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input type="radio"/>	BELEGE[1]	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input type="radio"/>	ADRESSEN[1]	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	ARTIKEL.UPDATE[1]	Edit	Delete	Sync	Sync	Async	Async	Abrufen			

Funktionen

Im Bereich Funktionen können Sie auch mit dem PLUS-Symbol eigene Funktionen anlegen.

Active	Function			Execute	ExecExtra	Execute	ExecExtra			

Funktionen

Sie haben die Möglichkeit eine Datenressource (ARTIKEL, BELEG, usw.) oder direkt eine Funktion für eine Datenressource (ARTIKEL.GET, usw.) anzugeben.

Active	Function			Exec	ExecExtra	Exec	ExecExtra	Options			
<input type="radio"/>	ARTIKEL.GET[1]	Edit	Delete	Sync	Sync	Async	Async	Abrufen			
<input type="radio"/>	ARTIKEL[1]	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	ADRESSE[1]	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE

Funktionen

Sie können dann die angelegte Funktion mit den Funktionen nach Funktionsbeschreibung bearbeiten und Ausführen.

- EDIT Editieren der Funktionsbeschreibung
- DELETE Löschen der Funktionsbeschreibung
- SYNC Synchrone Ausführung der Funktion (jeweils einfach und mit Extra Parametern)
- ASYNC Asynchrone Ausführung Funktion (jeweils einfach und mit Extra Parametern)
- INSERT Ausführung der INSERT Funktion einer Datenressource (ARTIKEL.INSERT)
- UPDATE Ausführung UPDATE Funktion einer Datenressource (ARTIKEL.UPDATE)
- DELETE Ausführung DELETE Funktion einer Datenressource (ARTIKEL.DELETE)

WWSVC Funktion anlegen

Funktions-Name  
GET\_ARTIKEL

Version [optional]

# Name + + + + + Datentype

01	ARTNR	NUMBER
02		
03		
04		
05		
06		
07		
08		
09		

Funktion speichern

WWSVC Funktion aufrufen

Funktions-Name  
GET\_ARTIKEL

# Name + + + Wert

01 ARTNR

Funktion aufrufen

## WEBWARE iWWSVC.js

Beschreibung des Interfaces für WWSVC Javascript in der Version 1.10

### Auslieferung

Sie finden die aktuelle Version des iWWSVC.js jeweils im aktuellen Setup/Update der WEBWARE 2.01 und 2.02 im folgenden Verzeichnis

BIN\HOME\WWSDKS\WWSVC-JS\

Diese Dokumentation bzw. immer die aktuellste finden Sie unter

BIN\HOME\WWSDKS\WWSVC-JS\WWSVC-JS.pdf

Ebenso in der Yammer DevCamp:WWSVC Gruppe auf der rechten Seite unter Netzwerkressourcen

### Zielbeschreibung

Das hier vorliegende Javascript Interface soll Ihnen helfen die WWSVC Schnittstelle der WEBWARE WEB Services (WWSVC) zu Verstehen und im Browser anzuwenden, bzw. für Ihr Zielsystem entsprechend zu implementieren.

Das iWWSVC.js speichert dabei notwendige Daten im Local-Storage des Browser um diese auch nach einem Browserneustart für Sie zur Verfügung zu haben.

Neben Hersteller-/Anwendungen, Service-Pässen, Session-Token, Asynchron-Abfragen haben Sie auch die Möglichkeit einfache Funktionen abzubilden und mit unterschiedlichen Aktionen auszuführen.

### Aufbau der iWWSVC-JS Schnittstelle

Das Interface besteht hierbei aus 3 Schichten:

- Oberfläche WWSVC-js.html/svcjs.css/testapp.js

Neben der HTML/CSS Datei für die Oberflächen Gestaltung finden Sie in der Datei TESTAPP.JS die Anwendung der iWWSVC-js Schnittstelle als Beispiel.

- iWWSVC-js WW-SVC-BASE.js

In dieser Datei sind die aktuellen Funktionen der iWWSVC-js dokumentiert sowie implementiert.

- Interne WEBWARE Funktionen WW-SVC-MIN.JS bzw. WW-SVC-INT.JS

In der Datei WW-SVC-BASE.js sind die Schnittstellenfunktionen enthalten die von Ihnen verwendet werden sollen. Dabei werden von der Datei WW-SVC-BASE.js die internen Funktionen aus der Datei WW-SVC-MIN.Js gekapselt und mit einem einfachen Interface versehen. Hier sind also in einer Datei alle Tool und Grundfunktionen zusammengefasst welche Low-Level für die Verwaltung von Service-Pässen, Hash-Wert Berechnung, Kommunikationsparameter Erstellung usw. notwendig sind.



## iWWSVC-js Test-Anwendung / Oberfläche

Verwaltungs-Funktionen der Test-Anwendung

- mehrere Anwendungen
- je ein Service-Pass je Anwendung
- je ein aktueller Session-Token je Anwendung
- Eingabe von mehreren Funktionen für die Ausführung

Die Oberfläche wird durch 3 Rahmen dargestellt, Anwendungen, ServicePass und Funktionen. Durch Klick auf das PLUS-Symbol können Sie jeweils neue Elemente entsprechend des Rahmens anlegen.

WWSVC Test Anwendung 1.03 / 21.02.2016

Active	Hersteller	Anwendung	Options		
<input type="radio"/>	53f69160a5b0b89136ba1c6390c1e5d1	04abf1c38b8522869f857dcffa3c5500	Edit	Delete	Abrufen
<input type="radio"/>	53f69160a5b0b89136ba1c6390c1e5d1	04abf1c38b8522869f857dcffa3c5500	Edit	Delete	Abrufen
<input checked="" type="radio"/>	da172c74e79c8886bc5aa87979e953cg	58e309af4f42dac98ece7b958fee0e0i	Edit	Delete	Abrufen

Anwendungen

Service-Pass	Session-Token	Pass	Pass	Session	Options
c1d9136edf6424c828ae2d282ec262cb					Validate DeRegister Anfordern Abrufen

ServicePass

Active	Function	Exec	ExecExtra	Exec	ExecExtra	Options
<input type="radio"/>	ARTIKEL[1]	Edit	Delete	Sync	Sync	Async Async Abrufen INSERT UPDATE DELETE
<input type="radio"/>	ADRESSE[1]	Edit	Delete	Sync	Sync	Async Async Abrufen INSERT UPDATE DELETE

Funktionen

Aktionen für bestehende Objekte können Sie innerhalb der Tabellendarstellung entsprechend auslösen. Beispiel Validieren eines Service-Passes durch Klick auf die Schaltfläche "Validate" in der Service-Pass Zeile, oder durch Session Anfordern ein Session-Token für einen Service-Pass anfordern.

In der Datei TESTAPP.JS finden Sie entsprechende Call-Back-Funktionen welche durch Klick auf die Oberflächen-Controls ausgelöst werden.

Ein großer Teil der Test-Anwendung kümmert sich um die Speicherung und Ausführung von Funktionen welche innerhalb der iWWSVC-js nicht benötigt werden.



## iWWSVC-js Basis-Schnittstelle

Die Basis-Schnittstelle deckt mit 16 Grundfunktionen alle Bereiche des WEBWARE WEB Service (WWSVC) Systems ab.

Die Funktionen sowie die Ein- und Ausgangsparameter werden in der Datei WW-SVC-BASE.js genauer beschrieben. Die Beschreibung hier soll Ihnen einen kurzen Überblick geben.

### Grundobjekte / Begriffe

AppHndl	Zugriffsschlüssel auf eine interne Struktur mit Informationen zu einer Anwendung
ServicePass	Verwaltungsfunktion für einen Zugriffsschlüssel einer Anwendung
SessionToken	Zugriffserlaubnis die zeitlich begrenzt und per Benutzername / Passwort gesichert ist

### Einleitung Beschreibung der Basis-Schnittstelle

Das hier verfügbare Javascript Interface für die WWSVC (iWWSVC-JS) speichert für Sie die notwendigen Zustände und Laufzeitdaten.

Der Ablauf bzw. die Anwendung wird im folgenden beschrieben:

## **A) Anwendungs-Verwaltung**

Verfügbare Funktionen:

WWSVC\_CONFIGURE\_WW\_SERVICE\_APP (...)  
WWSVC\_GET\_WW\_SERVICE\_APP\_HANDLE (...)  
WWSVC\_GET\_APPINFO (...)  
WWSVC\_DELETE\_AI (...)  
WWSVC\_GET\_APPLICATION\_OPTIONS (...)

### **Anlagen bzw. Ändern einer Anwendungsbeschreibung**

Rufen Sie für die Anlage bzw. zum Ändern von Anwendungsparametern zuerst diese Funktion:

```
WWSVC_CONFIGURE_WW_SERVICE_APP (WWSVC_SERVER_URL, WWSVC_SERVER_PORT,  
HERSTELLERID, ANWENDUNGSID, APP_SECURE_ID, APP_REVISION)
```

auf und übergeben Sie hier die notwendigen Informationen. Die iWWSVC-JS speichert diese und gibt einen Anwendungs-Handle AppHndl zurück.

Wird die Funktion mit den Daten einer bestehenden Anwendung aufgerufen werden die Parameter aktualisiert. Es können mehrere Anwendungen in der iWWSVC-JS gleichzeitig gespeichert werden.

Falls Sie für eine bestehende Anwendung den Anwendungs-Handle, Informationen oder die Anwendung löschen wollen stehen hierzu die 3 übrigen Funktionen zur Verfügung:

**Hole einen Anwendungs-Handle (AppHndl)**

Hiermit können Sie einen AppHndl für eine bestehende Anwendung abfragen:

```
WWSVC_GET_WW_SERVICE_APP_HANDLE (HERSTELLERID, ANWENDUNGSID,
                                   APP_SECURE_ID, APP_REVISION)
```

**Gebe die Beschreibung für einen AppHndl**

Übergabe eines AppHndl an die Funktion, und man erhält eine Struktur mit den internen Beschreibungen

```
WWSVC_GET_APPINFO (AppHndl)
```

**Lösche die Anwendungs-Informationen für einen AppHndl**

Mit der Übergabe eines AppHndl an diese Funktion wird die interne Struktur für eine Anwendung gelöscht

```
WWSVC_DELETE_AI (AppHndl)
```

**Abrufen der aktuellen WWSVC-Schnittstelle**

Mit der Übergabe eines AppHndl an diese Funktion wird bei dem Zugehörigen WWSVC-Server die JSON-API-Beschreibung für die ausführbaren Funktionen abgerufen. Sie erhalten dabei ein JSON-Objekt mit Funktionen und deren Eingangs und Ausgangsparametern.

```
WWSVC_GET_APPLICATION_OPTIONS (AppHndl, aCB)
```

**B) Verwaltung von Service-Pässen**

Für den Zugriff auf die WWSVC benötigen Sie neben der Anwendungs-Information einen ServicePass der den Zugriff auf die WEBWARE SVC absichert. Hierzu stehen Ihnen 3 Funktionen zur Verfügung die es erlauben für eine Anwendung einen ServicePass zu

**Registrieren**

bei Vorgabe von WWSVC-Server auch mit Angabe von Benutzer/Passwort

```
WWSVC_SERVICEPASS_REGISTER (AppHndl, aCB, opt_User, opt_Password)
```

**Prüfen**

ob der Service-Pass Erlaubt ist, bzw. mittlerweile freigegeben wurde

```
WWSVC_SERVICEPASS_VALIDATE (AppHndl, aCB)
```

**Löschen/entfernen**

also den Service-Pass beim WW-Server zu entfernen und dann im Browser die interne Service-Pass Struktur zu Löschen

```
WWSVC_SERVICEPASS_REMOVE (AppHndl, aCB)
```

**C) Anforderung eines Session-Token**

Je nach Vorgabe im WWSVC-Servers kann die zusätzliche Authentifizierung mit Hilfe eines Session-Token erforderlich sein. Hier stehen Ihnen 2 Funktionen zur Verfügung

**Neuer Session-Token anfordern**

```
WWSVC_CONNECT_SESSION (AppHndl, UsrNam, UsrPW, aCB)
```

**Vorhandener Session-Token löschen**

```
WWSVC_DISCONNECT_SESSION (AppHndl, aCB)
```

**D) Ausführen von WWSVC Funktionen**

Für die Ausführung einer Funktion ist die Voraussetzung das Sie einen gültigen ServicePass für den Zugriff auf den Ziel-WWSVC Server haben. Falls als erweiterte Anforderung die Benutzung von Session-Token erforderlich ist, so muss dieser ebenfalls vorhanden sein.

Es gibt 2 Funktionen die Sie für die Ausführung der WWSVC-Funktionen mit iWWSVC-JS für den direkten Aufruf verwenden können:

**1) Aufruf mit Einzel-Angabe von Parametern**

Hierbei werden die Funktionsparameter einzeln übergeben.

```
WWSVC_CALL_FUNCTION (AppHndl, FuncName, ParamValues, aCB,
                    opt_FuncVer, opt_Async, opt_VariableNameArray,
                    opt_VariableTypeArray, opt_AsyncPollInSecond,
                    opt_AsynchMaxTimeToPoll)
```

**2) Aufruf mit Hilfs-Funktionen**

Um die Funktionsparameter sowie Synchron/Asynchron Parameter korrekt übergeben zu können gibt es 2 Hilfs-Funktionen mit denen diese Entsprechend formatiert werden:

**Erstelle Funktionsparameter**

```
WWSVC_HELPER_BUILD_FUNC_PARAM (FuncName, ParamValueArray, opt_FuncVer,
                               opt_ParamNameArray, opt_ParamTypeArray)
```

**Erstelle Asynchron-Parameter**

```
WWSVC_HELPER_BUILD_ASYNC_PARAM (ExecAsync, AsyncPollInSecond,
                                AsynchMaxTimeToPoll)
```

**Die Ausführung erfolgt dann mit der Funktion**

```
WWSVC_CALL_SVC_FUNCTION (AppHndl, aCB, FuncParam, opt_AsyncParam)
```

in dem Sie an diese die Ergebnisse der Hilfs-Funktionen sowie AppHndl und aCB() übergeben.

**3) Abruf von Optionen für Datenressourcen/Funktionen**

Wenn Sie an diese Funktion einen AppHndl sowie einen Funktionsnamen übergeben, so erhalten Sie für das Objekt eine JSON-API Struktur mit der Beschreibung der Funktion und deren Parameter.

```
WWSVC_GET_FUNCTION_OPTIONS (AppHndl, FuncName, aCB, opt_FuncVer)
```

Wenn Sie an diese Funktion einen AppHndl sowie einen leeren Funktionsnamen übergeben, so erhalten Sie für die Liste der Ressourcen für die Anwendung übergeben.

```
WWSVC_GET_FUNCTION_OPTIONS (AppHndl, null, aCB, opt_FuncVer)
```

Wenn Sie an diese Funktion anstatt des Funktionsnamen den Namen einer Ressource übergeben erhalten Sie hier die Liste der Funktionen für die Resource

```
WWSVC_GET_FUNCTION_OPTIONS (AppHndl, "BELEGE", aCB, opt_FuncVer)
```

## E) Import Funktion

Mit Hilfe der Funktion IMPORT.EXEC können Dateien an die Import-Funktion des WWSVC übergeben werden. Dabei wird die EXECURL Schnittstelle verwendet. Hierzu steht Ihnen die Funktion

```
WWSVC_SEND_IMPORT_FILE (AppHndl, aCB, aExecMode, aFileObject,  
                        opt_FuncName, opt_FuncVer, opt_Async)
```

zur Verfügung. Ein Beispiel für die Verwendung finden Sie in TESTAPP.js

## F) Beschreibung der erwarteten Callback-Funktion

Da die meisten Aktionen Asynchron ausgeführt werden, d. h. das Programm wartet auf die Antwort des WWSVC-Servers würde die Anwendung bis zum Eintreffen der Antwort "einfrieren". Daher werden bei einigen Funktionsaufrufen sogenannte Call-Back-Funktionen (Rück-Ruf-Funktionen) mit übergeben, welche nach Eintreffen des Ergebnisses ausgeführt werden.

Bis auf die Hilfsfunktionen zum Füllen von Funktionsparametern werden bei allen iWWSVC-js-Funktionen die Übergabe von Call-Back Funktionen erwartet. Wird diese nicht übergeben so werden die Aktionen trotzdem durchgeführt, jedoch erhält man dann keine Rückmeldung über den Erfolg bzw. die Daten von ausgelösten Funktionen

Hier die Beschreibung des Parameters aCB der an diese iWWSVC-sj Funktionen übergeben wird. Es ist zu Beachten das je nach Funktionsaufruf nicht alle Felder gefüllt werden.

aCB : Callback-Funktion mit Parametern CBP welcher aus folgenden Einträgen besteht

CBP.RESULTCODE	0 Fehler
	1 OK
CBP.ERRINFO	Beschreibung Rückgabe-Code/Zustand/Fehlerinfo
CBP.HTTPSTATUS	HTTP Status-Code
CBP.HEAD	COMESULT Objekt das zurückgegeben wurde
CBP.OBJ	Komplettes JSON Objekt aus der Rückgabe
CBP.BODY	Original HTTP-Body der empfangen wurde
CBP.MODE	Hier wird 'R' für Registrierung gesetzt

Um die Verwendung von Call-Back-Funktionen zu Verstehen sehen Sie diese angewendet in der TESTAPP.JS.

## WEBWARE SVC webServices

### Wie sieht ein Zugriff auf WWSVC aus

#### Folgende Aufgaben sind für einen Erstzugriff notwendig

- Administrator gibt eine Hersteller/App als SecuredApp für eine WW-Server-Instanz frei
- Client Registriert sich über den WWSVC-ServicePunkt für die SecuredApp
- WW-Server akzeptiert die Registrierung und übergibt PASS-ID und APP-Secret
- Automatische Freigabe, oder WW-Server Administrator gibt gegeben falls den Zugriff manuell frei
- Client speichert lokal PASS-ID und APP-Secret
- Start Anwendungs-Zugriff....

#### Notwendige Aufgaben für einen registrierten Client

- Client Sendet EXEC Anfrage an WWSVC-ServicePunkt mit PASS-ID und Funktionsanfrage
- Client Sendet bei Bedarf eine CONNECT-Anforderung falls Zugriff mit Benutzer/Passwort aktiviert ist.
- WEBWARE-Server bearbeitet die Anfrage und sendet Antwort

#### Zugriffsmodus Synchron/Asynchron

In der Basiskonfiguration der WEBWARE Services werden alle Anfragen SYNCHRON ausgeführt. Dabei wird die Verbindung beim Start einer Service-Funktion solange beim WEBWARE-Server gehalten bis ein Ergebnis für die Service-Funktion vorliegt.

Um Ihnen und damit Ihren Anwendungen bei der Verwendung von Service-Funktionen mehr Flexibilität zu verschaffen, ist es auch möglich Service-Funktionen Asynchron abzusetzen, wobei dann die Verbindung direkt nach der erfolgreichen Einlastung der Service-Funktion mit einem zugehörigen Asynchron-Handle zurückgegeben wird. Es ist dann möglich zu einem späteren Zeitpunkt mit Hilfe des Asynchron-Handle das Ergebnis der Service-Funktion abzurufen. Desweiteren steht ein weiterer Modus zur Verfügung mit dem kein Abrufen des Asynchronen Ergebnisses vorgesehen ist.

#### /WWSVC der ServicePunkt

Der Zugriff auf den Service-Zugang der WEBWARE erfolgt über einen definierten Service-Punkt mit der Kennung WWSVC. Alle Anfragen die über diesen Service-Punkt erfolgen werden in der Folge als WEBSERVICE-Anfrage interpretiert.

Der Service-Punkt bildet sich aus der WEBSERVER-Domain sowie /WWSVC. Da der WEBWARE-Server mehrere Internet-Zugangspunkte für mehrere WEBWARE-Instanzen anbietet, stehen die Service-Punkte damit für jede Instanz zur Verfügung.

Beispiel: <https://MeineDomain.de/WWSVC>

<https://MeineDomain.de:4443/WWSVC>

## **/WWSVC/WWSERVICE Funktionen**

### **Schnittstellenbeschreibung**

Der /WWSVC-ServicePunkt bietet unterschiedliche Grundfunktionen an, mit denen ein Client mit dem WEBWARE-Server kommunizieren kann. Durch Angabe der WWSERVICE-Funktions-Kennung wird die Kommunikation direkt mit dem WW-Server aufgenommen.

Hier stehen Funktionen für die Registrierung, Prüfung, Löschung, Validierung eines Service-Passes und Verbindungs-Ende zur Verfügung. Dabei ist es teilweise notwendig weitere Funktionsparameter bzw. Cookies mit zu geben.

### **REGISTER**

Ein Endgerät registriert sich als Client beim WEBWARE-Server für eine Secured-App. Hierbei wird Hersteller/Anwendung und Optional Benutzer/Passwort mit gegeben. Ist die Registrierung erfolgreich wird ein Service-Pass ausgestellt und zurückgegeben. Der Service-Pass ist nur dann sofort gültig wenn keine Freigabe durch den WEBWARE-Administrator notwendig ist.

### **VALIDATE**

Prüfe ob ein Service-Pass für den Zugriff verwendet werden kann. Hierzu wird der Service-Pass übergeben und der Aufrufer erhält die Info ob der Service-Pass gültig ist.

### **DERGISTER**

Wird ein Service-Pass nicht mehr benötigt, so kann mit der DERIGSTER-Funktion ein Service-Pass entfernt werden.

### **CONNECT**

Je nach Vorgabe des WEBWARE-Administrator für eine Secured-App kann es erforderlich sein, das bei einem Zugriff mit einem Service-Pass eine Authentifizierung, also die Anmeldung mit Benutzer und Passwort notwendig ist. Hierzu dient die Connect-Funktion. Bei dieser wird neben dem Service-Pass auch Benutzer und Passwort übergeben, welches vom WEBWARE-Server anhand von Benutzer-Listen für den Service-Pass ausgewertet werden.

Ist der Zugriff erlaubt, so wird ein Session-Token erstellt, welcher den Zugriff für eine bestimmte Zeit auf den Service-Pass erlaubt und der bei jeder Anfrage mit gegeben werden muss.

Erfolgt ein Zugriff auf einen Service-Pass ohne gültigen Session-Token, auch Erst-Zugriff, so wird die Fehlermeldung / Fehlercode 401 Authorization required (Anmeldung erforderlich)

### **CLOSE**

Mit der CLOSE-Funktion kann ein Session-Token entwertet werden, also die Zugangserlaubnis für diesen Service-Pass aufgehoben werden. Ebenso wird damit der Zustand aktiv im System aufgehoben.

### **CURSORCLOSE**

Mit der CURSORCLOSE Funktion kann ein bestehender Cursor manuell vom Client aus Geschlossen werden. Nach dieser Funktion ist der WWSVC-CURSOR nicht mehr gültig.

### **GETASYNCRESULT**

Mit dem GetAsyncResult-Zugriff kann das Ergebnis für eine asynchron aufgegebene Service-Funktion abgeholt werden. Um eine Service-Funktion Asynchron abarbeiten zu lassen, muss diese eine Cookie mit der Kennung WWSVC-EXECUTE-MODE: ASYNCHRON, bzw. in der JSON-Struktur für den Service-pass die Variable "EXECUTE\_MODE":"ASYNCHRON" setzen. Näheres hierzu weiter unten.

## OPTIONS

Mit dem OPTIONS Befehl können Sie eine JSON-API-Beschreibung aller verfügbaren Schnittstellen Funktionen abrufen. Diese wird aus der Datei bin\wws\wwsvc\WWSVC-JSON.API gelesen.

## WEBPROXY

Mit dem Zugangspunkt WEBPROXY kann ein HTTPS Proxy verwendet werden, der es erlaubt Zugriffe auf Fremde WEB-Server durchzuführen. Dies wird dann notwendig wenn beim Zugriff aus dem Browser der Verbindungsaufbau wegen CORS - Einschränkungen nicht möglich ist.

## Globale Rückgabe Information: COMRESULT

Wird ein Service-Pass bearbeitet und die Rückgabe ist keine binäre Datei, so erhält man ein JSON-Objekt mit Informationen über den Verarbeitungszustand und das Ergebnis zurück. Die COMRESULT Struktur hat folgende Felder:

- COMRESULT
  - STATUS                      Entspricht dem HTTP-Status-Code (200=OK)
  - CODE                        Entspricht dem HTTP-Status-Info ("200 OK")

Es folgen Optionale Felder, die teilweise über System-Wert (Detail-Level) aktiviert werden können.

- [INFO                        Text-Information über die Verarbeitung]
- [ERRORCODE                Fehler/Hinweis-Code für das Ergebnis]
- [ERRORLINK                WEB-Link auf die Dokumentation zu dem Error-Code]
- [ERRORINFO                Genauere Fehler-Informationen, zur Fehleranalyse]
- [WWSVC\_ASYNCHRON\_HANDLE]    Asynchroner Service-Handle bei Asynchron..

Werte in []-Klammern sind Optional

Beispiel:

```
{
  "COMRESULT":
  {
    "STATUS": 406,
    "CODE": "406 Not Acceptable",
    "INFO": "REGISTER is forbidden",
    "ERRORCODE": 10001,
    "ERRORLINK": "DOCWWSVC/ERR.HTML/#01001",
    "ERROINFO": "WWSVC IS NOT ACTIVATED, CALL ADMINISATRATOR"
  }
}
```

## WWSVC/WWSERVICE - Funktionen im Detail

### Registrierung eines Client /WWSVC/WWSERVICE/REGISTER

Ein Endgerät registriert sich als Client beim WEBWARE-Server für eine Secured-App. Hierbei wird Hersteller/Anwendung und Optional Benutzer/Passwort mit gegeben. Ist die Registrierung erfolgreich wird ein Service-Pass ausgestellt und zurückgegeben. Der Service-Pass ist nur dann sofort gültig wenn keine Freigabe durch den WEBWARE-Administrator notwendig ist.

ServicePunkt: `https://MeineDomain.de/WWSVC/WWSERVICE/REGISTER/`

Möchte sich ein Client neu bei einem WW-Server registrieren so muss er hier folgende Parameter mit übergeben.

- **HESTELLER**                      Hersteller-ID 32 Byte HASH-Wert
- **ANWENDUNG**                    Anwendungs-ID 32 Byte HASH-Wert
- **SecureAppID**                    Angabe der internen Secure-AppID
- **Revision**                        Optional Revision der Anwendung
- **USER**                            Optional Benutzer Name
- **PASSWORD**                      Optional Benutzer Passwort
- **CLIENTINFO**                    Beschreibung des Client, Text für Administrator (wer ist es)
- **CLIENTSECRET**                Optional, je nach Client Referenz-Hardware-Informationen

Beispiel eines Aufrufs

`../WWSVC/WWSERVICE/REGISTER/HERSTELLER/APPID/SecureID//S.MUELLER//Client../`

und nun eine komplette HTTP-Anfrage

```
GET /WWSVC/WWSERVICE/REGISTER/53f69160a5b0b89136ba1c6390c1e5d1/04abf1c38b8522869f857deffa3c5500/1//Test-User// HTTP/1.1
Host: myDomain.de
Connection: keep-alive
```

Es werden keine Cookies benötigt.

Der WEBWARE-Server prüft dann ob eine SecuredApp mit Hersteller-ID und Anwendungs-ID sowie Secure-App-ID für diesen ServicePunkt verfügbar ist und wie der Zugriff darauf erlaubt ist.

Wird vom Administrator eine Benutzer-Liste für die Registrierung vorgegeben, so wird der Benutzer-Name sowie das Passwort gegen diese Benutzer-Liste geprüft.

Der Rückgabe-Code sowie die Rückgabe-Daten hängen dabei von der Prüfung der Registrierung sowie des Freigabe-Modus (Automatisch bzw. manuelle Freigabe) für die Secured-App ab.



**Mögliche Rückgabe-Codes**

- 404 WWSCVC Subsystem ist auf dem WW-Server nicht verfügbar
- 406 Register für Anwendung nicht möglich, meist Anwendung nicht bekannt
- 202 Registrierung ok, jedoch Freigabe von Admin ausstehend
- 200 Registrierung ok, es kann sofort mit dem Service-Pass gearbeitet werden

**Registrierung abgelehnt**

Ist die SecuredApp (HerstellerID/AnwendungID) nicht vorhanden, oder aktuell für Registrierungen gesperrt, bzw. Benutzer/Passwort nicht in einer vorhandenen Benutzer-Liste enthalten, so erhält der Client den HTTP-Statuscode 406 Not Acceptable als Antwort, sowie ein JSON-Objekt mit weiteren Informationen.

HTTP/1.1 406 Not Acceptable  
Content-Length: 94  
Content-Type: application/json

Date: Tue, 17 Mar 2015 23:37:33 GMT  
Server: WebWare PoWeR Server

```
{ "COMRESULT":  
  { "STATUS": 406, "CODE": "406 Not Acceptable", "INFO": "REGISTER is forbidden",  
    "ERRORCODE": 10001, "ERRORINFO": "WWSVCDOKU/ERR.HTML/#01001"} }  
{ "COMRESULT":  
  {  
    "STATUS": 406,  
    "CODE": "406 Not Acceptable",  
    "INFO": "REGISTER is not possible",  
    "ERRORCODE": 50100,  
    "ERRORLINK": "DOCWWSVC/ERR.HTML/#50100",  
    "ERRORINFO": "APPLICATION NOT KNOWN"  
  }  
}
```

**Registrierung OK, Freigabe von Administrator ausstehend**

Ist die SecuredApp (HerstellerID/AnwendungID) vorhanden, aber der Administrator hat diese auf "Freigabe durch Administrator" konfiguriert, so erhält der Client bereits den Service-Pass mit dem Hinweis das der Zugriff noch nicht möglich ist. Erst durch Freigabe durch den Administrator ist der Zugang möglich. Wird in der Folge mit diesem Service-Pass

Hierbei wird der Return-Code HTTP 202 Accepted zurückgegeben, mit dem Service-Pass als JSON-Objekt im Datenbereich.

HTTP/1.1 202 Accepted  
Content-Length: 246  
Content-Type: text/html  
Date: Tue, 17 Mar 2015 23:41:05 GMT  
Server: WebWare PoWeR Server

```
{ "COMRESULT":  
  { "STATUS": 202,  
    "CODE": "202 Accepted",  
    "INFO": "REGISTER OK, WAIT FOR ADMIN RELEASE",  
    "ERRORCODE": 10000,  
  }  
}
```

```
"ERRORLINK": "DOCWWSVC/INFO.HTML/#10000",
"ERRORINFO": "REGISTER OK WAIT FOR ADMIN RELEASE"
},
"SERVICEPASS":
{
  "PASSID": "9ff37b3b234b45d811acdc50fd0f80da",
  "APPID": "cb530381bbbb18cd33923c433ef7ee5c",
  "PDATE": 20150622,
  "PTIME": 8190176 }
}
```

### Registrierung OK

Ist die SecuredApp (HerstellerID/AnwendungID) vorhanden und mit automatischer Freigabe konfiguriert, so ist nach der Registrierung der sofortige Zugriff mit dem Service-Pass möglich.

Hierbei wird der Return-Code HTTP 200 OK zurückgegeben, mit dem Service-Pass als JSON-Objekt im Datenbereich.

```
HTTP/1.1 200 OK
Content-Length: 216
Content-Type: text/html
Date: Tue, 17 Mar 2015 23:39:13 GMT
Server: WebWare PoWeR Server
```

```
{ "COMRESULT": {
  "STATUS": 200,
  "CODE": "200 OK",
  "INFO": "REGISTER OK"
  "ERRORCODE": 0,
},
"SERVICEPASS":
{
  "PASSID": "44305f615eadfaca901b60692eed7f4",
  "APPID": "38af61d2aff033ece56ba16ae0cbf472",
  "PDATE": 20150318,
  "PTIME": 391374
}}
}}
```

Der Client muss dann die Daten des Service Passes lokal speichern, so dass er diese beim Zugriff auf den WW-SVC ServicePunkt übergeben kann.

PASSID: Service-Pass ID für Zugriff auf den WWSVC-Service-Punkt

APPID: APP-SECRET Application Secret, für Bildung von Dynamischen HASH-Werten

PDATE+PTIME sind die Erzeugungs-Datum/Uhrzeit auf Seiten des WW-Servers. Diese werden später benötigt um dynamische Hash-Werte zu Erzeugen.

**Validierung ServicePass: /WWSVC/WWSERVICE/VALIDATE**

ServicePunkt: <https://MeineDomain.de/WWSVC/WWSERVICE/VALIDATE/ServicePassID>

Hat ein Client einen ServicePass, und möchte prüfen ob dieser gültig ist, so kann er die VALIDATE-Funktion verwenden. Hierbei müssen neben dem Service-Pass auch die korrekten Validierungs-Cookie gesetzt werden.

Beispiel eines Aufrufs

../WWSVC/WWSERVICE/VALIDATE/123abcdServicePassIDrstruvxyz789

und nun eine komplette HTTP-Anfrage

GET /WWSVC/WWSERVICE/VALIDATE/0778e54d57f23b58bb0cdb2cb1e45691 HTTP/1.1

Connection: keep-alive

Pragma: no-cache

Cache-Control: no-cache

WWSVC-HASH: 88e49c05f9c49d86a2465117ec7e9de1

WWSVC-REQID: 14

WWSVC-TS: Tue, 17 Mar 2015 23:33:57 GMT

**Mögliche Rückgabe-Codes**

- 404 WWSVC Subsystem ist auf dem WW-Server nicht verfügbar bzw.
- 404 Service-Pass nicht bekannt
- 202 Service-Pass ist OK, jedoch Freigabe von Admin ausstehend
- 200 Service-Pass ist OK, es kann sofort mit dem Service-Pass gearbeitet werden

**Validierung schlägt fehl**

Die Validierung des ServicePasses schlägt fehl wenn einer dieser Punkte zutrifft:

- ServicePass ist nicht bekannt
- ServicePass ist gesperrt
- WWSVC-APPID passt nicht zum ServicePass
- Zugriff ist wegen Zeitvorgaben aktuell nicht möglich
- Zugriff ist wegen Netzbeschränkungen nicht möglich
- ServicePass ist abgelaufen

Es wird dann mit dem HTTP-Code "404 Ressource not found" geantwortet. Im Body wird dazu ein JSON-Objekt mit COMRESULT-Objekt zurückgegeben welche genauere Informationen über den Grund der Ablehnung enthält.

HTTP/1.1 404 Resource not found  
Content-Length: 104  
Content-Type: text/html  
Date: Tue, 17 Mar 2015 23:53:46 GMT  
Server: WebWare PoWeR Server

```
{ "COMRESULT":  
  {  
    "STATUS": 404,  
    "CODE": "404 Resource not found",  
    "INFO": "ERROR ServicePass not known",  
    "ERRORCODE": 1002,  
    "ERRORLINK": "WWSVCDOKU/ERR.HTML/#01002"  
  }  
}
```

### Validierung OK

Wird der ServicePass als gültig erkannt so wird der HTTP-Code 200 OK zurückgegeben. Im Body der HTTP-Antwort wird dann ein JSON COMRESULT-Objekt mit den Standardinformationen für Code 200 zurückgegeben.

Beispiel für HTTP Rückgabe bei Validierung (Service-Pass nicht freigegeben Code 202)

HTTP/1.1 202 Accepted  
Content-Length: 98  
Content-Type: text/html  
Date: Tue, 17 Mar 2015 23:50:16 GMT  
Server: WebWare PoWeR Server

```
{ "COMRESULT":  
  {  
    "STATUS": 202,  
    "CODE": "202 Accepted",  
    "INFO": "SERVICEPASS WAITING FOR RELEASE"  
  }  
}
```

Beispiel für HTTP Rückgabe bei Validierung (Service-Pass ist freigegeben Code 200)

HTTP/1.1 200 OK  
Content-Length: 75  
Content-Type: text/html  
Date: Tue, 17 Mar 2015 23:52:03 GMT  
Server: WebWare PoWeR Server

```
{ "COMRESULT":  
  {  
    "STATUS": 200,  
    "CODE": "200 OK",  
    "INFO": "SERVICEPASS OK"  
  }  
}
```

## **De-Registrierung eines Client /WWSVC/WWSERVICE/DEREGISTER**

ServicePunkt: `https://MeineDomain.de/WWSVC/WWSERVICE/DEREGISTER/ServicePass`

Möchte sich ein Client seinen ServicePass löschen, so kann er mit dieser Funktion den ServicePass für den Zugriff sperren lassen.

Diese Funktion macht dann Sinn wenn ein Client die Zugriffssoftware löscht, bzw. er den ServicePass für die SecuredApp nicht mehr benötigt.

`../WWSVC/WWSERVICE/DEREGISTER/123abcdServicePassIDrstruvxyz789`

und nun eine komplette HTTP-Anfrage

```
GET /WWSVC/WWSERVICE/DEREGISTER/3e16a4e8061f8122bb782c2c1a340f2f HTTP/1.1
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
WWSVC-HASH: b866ec8edafbd50a4bd9c506afc99cfa
WWSVC-REQID: 38
WWSVC-TS: Tue, 17 Mar 2015 23:53:42 GMT
```

### **Mögliche Rückgabe-Codes**

- 404 WWSVC Subsystem ist auf dem WW-Server nicht verfügbar bzw.
- 404 Service-Pass nicht bekannt
- 200 Service-Pass wurde erfolgreich entfernt

**DeRegistrierung schlägt fehl**

Die DeRegistrierung des ServicePasses schlägt fehl, wenn einer dieser Punkte zutrifft:

- ServicePass ist nicht bekannt
- WWSVC-APPID passt nicht zum ServicePass

Es wird dann mit dem HTTP-Code 405 Not Allowed geantwortet. Im Body wird dazu ein JSON-Objekt mit COMRESULT-Objekt zurückgegeben welche genauere Informationen über den Grund der Ablehnung enthält.

HTTP/1.1 404 Resource not found  
Content-Length: 104  
Content-Type: text/html  
Date: Tue, 17 Mar 2015 23:57:29 GMT  
Server: WebWare PoWeR Server

```
{ "COMRESULT":  
  {  
    "STATUS": 404,  
    "CODE": "404 Resource not found",  
    "INFO": "ERROR ServicePass not known",  
    "ERRORCODE": 50200,  
    "ERRORLINK": "DOCWWSVC/ERR.HTML/#50200",  
    "ERRORINFO": ""  
  }  
}
```

**DeRegistrierung OK**

Wird der ServicePass als gültig erkannt so wird der HTTP-Code 200 OK zurückgegeben. Im Body wird dann ein JSON COMRESULT-Objekt mit den Standardinformationen für Code 200 zurückgegeben. Der WW SVC löscht dann den ServicePass aus der Zugriffsliste, so dass keine weitere Kommunikation mit diesem Pass möglich ist.

HTTP/1.1 200 OK  
Content-Length: 85  
Content-Type: text/html  
Date: Tue, 17 Mar 2015 23:53:42 GMT  
Server: WebWare PoWeR Server

```
{ "COMRESULT":  
  {  
    "STATUS": 200,  
    "CODE": "200 OK",  
    "INFO": "SERVICEPASS DEREGISTERED"  
  }  
}
```

## **Neue Sitzung mit Benutzer-Liste eines Client /WWSVC/WWSERVICE/CONNECT**

Der Zugriff auf die WEBWARE Service-Schnittstelle mit einem gültigen Service-Pass kann weiter eingeschränkt werden. Dabei muss vor dem Zugriff auf die Service-Schnittstelle für einen Service-Pass eine Benutzer-Authentifizierung erfolgen.

Hierzu kann der Administrator in der SecuredApp eine gültige Benutzer/Passwort Liste hinterlegen und Vorgeben das bei einem Zugriff auf die SecuredApp der Client sich mit Benutzer und Passwort Anmelden muss.

Die Anmeldung erfolgt dabei über die CONNECT-Funktion bei der neben Service-Pass auch Benutzer und Passwort mit übergeben werden müssen.

Wurde die Anmeldung erfolgreich durchgeführt, wird ein Session-Token mit Hilfe eines Cookies: WWSVC-SESSION-TOKEN=xxx zurückgeben. Dieser WWSVC-SESSION-TOKEN Cookie muss dann bei allen weiteren Zugriffen mit den Service-Anfragen mit übergeben werden.

ServicePunkt:

`https://MeineDomain.de/WWSVC/WWSERVICE/CONNECT/ServicePass/Benutzer/Passwort`

Beispiel:

`/WWSVC/WWSERVICE/CONNECT/123abcdServicePassIDrstruvxyz789/Peter.Wolf/Entchen39`

`GET /WWSVC/WWSERVICE/CONNECT/...ServicePass.../Peter.Wolf/Entchen39`  
`Host: myDomain.de`

### **Mögliche Rückgabe-Codes**

- 406 Fehler kein gültiger ServicePass wurde übergeben
- 404 WWSVC Subsystem ist auf dem WW-Server nicht verfügbar bzw. ServicePass nicht bekannt
- 401 Fehler bei Benutzer/Passwort Prüfung bzw. Daten fehlen
- 200 Sitzung wurde erfolgreich gestartet

**Erfolgreiche Anmeldung:**

HTTP/1.1 200 OK

Content-Length: 85

Content-Type: text/html

Date: Tue, 17 Mar 2015 23:53:42 GMT

Server: WebWare PoWeR Server

Set-Cookie: WWSVC-SESSION-TOKEN ="04432570@1"; secure; httponly

```
{ "COMRESULT":
  {
    "STATUS": 200,
    "CODE": "200 OK",
    "INFO": "AUTHENTICATION OK "
  },
  {"SESSIONTOKEN":
    {
      "REQUIRED": 1,
      "WWSVC_SESSION_TOKEN": "...."
    }
  }
}
```

Bei weiteren Zugriffen muss dann der Session-Token mit geliefert werden. Hierzu gibt es die Möglichkeit den Session-Token als HTTP-Header

WWSVC-SESSION-TOKEN: "88e49c05f9c49d86a2465117ec7e9de1"

oder in der JSON-Struktur für den Service-Pass mit zu übergeben:

```
"WWSVC_PASSINFO":
{
  "SERVICEPASS": "0778e54d57f23b58bb0cdb2cb1e45691",
  "APPHASH ": "88e49c05f9c49d86a2465117ec7e9de1",
  "TIMESTAMP": "Tue, 17 Mar 2015 23:33:57 GMT",
  "REQUESTID": "42",
  "SESSION_TOKEN": "88e49c05f9c49d86a2465117ec7e9de1"
}, ....
```

**Ablehnung der Anmeldung:**ServicePass ungültig

Wird kein ServicePass übergeben der gültig ist so wird der Fehlercode 406 NOT ACCEPTABLE zurückgegeben.

HTTP/1.1 406 Not Acceptable

Content-Length: 85

Content-Type: application/json

Date: Tue, 17 Mar 2015 23:53:42 GMT

Server: WebWare PoWeR Server

```
{ "COMRESULT":
  {
    "STATUS": 406,
    "CODE": "406 Not Acceptable ",
    "INFO": " SESSIONTOKEN ERROR: NO VALID SERVICEPASS"
  }
}
```



ServicePass noch nicht freigegeben

Falls der Service-Pass noch nicht vom Administrator freigegeben ist, so wird dies mit der Fehlermeldung

```
{ "COMRESULT":  
  {  
    "STATUS": 406,  
    "CODE": "406 Not Acceptable ",  
    "INFO": "SESSIONTOKEN ERROR: SERVICEPASS WAIT FOR ADMIN RELEASE"  
  }  
}
```

Benutzer/Passwort Fehler

Wird kein passender Benutzer oder Passwort übergeben bzw. ist die Anmeldung nicht möglich so wird der Fehlercode 401 zurückgegeben.

HTTP/1.1 401 Authorization Required  
Content-Length: 85  
Content-Type: application/json  
Date: Tue, 17 Mar 2015 23:53:42 GMT  
Server: WebWare PoWeR Server

```
{ "COMRESULT":  
  {  
    "STATUS": 401,  
    "CODE": "401 Authorization Required ",  
    "INFO": "USER OR PASSWORD NOT KNOWN"  
  }  
}
```

## Beende die Verbindung eines Client /WWSVC/WWSERVICE/CLOSE

ServicePunkt: `https://MeineDomain.de/WWSVC/WWSERVICE/CLOSE/ServicePass`

Möchte ein Client die aktuelle Verbindung für einen ServicePass Beenden, so kann er dies mit der CLOSE-Funktion durchführen. Dabei wird auch ein etwaig vorhandener Session-Token (CONNECT-Funktion) widerrufen und ist dann nicht mehr gültig.

Es ist auch möglich die Verbindung "hart bzw. kalt" zu unterbrechen, jedoch können mit der CLOSE-Funktion, Ressourcen die der WW SVC verwendet sofort freigegeben werden.

`../WWSVC/WWSERVICE/CLOSE/123abcdServicePassIDrstruvxyz789`

und nun eine komplette HTTP-Anfrage

```
GET /WWSVC/WWSERVICE/CLOSE /123abcdServicePassIDrstruvxyz789
Host: myDomain.de
WWSVC-TS:
WWSVC-REQID:
WWSVC-APPID:
```

### Mögliche Rückgabe-Codes

- 406 Fehler kein gültiger ServicePass wurde übergeben
- 404 WWSVC Subsystem ist auf dem WW-Server nicht verfügbar bzw. ServicePass nicht bekannt
- 200 Sitzung wurde erfolgreich Beendet

### CLOSE OK

Der WW SVC Beendet alle Verbindungen für diesen ServicePass und gibt HTTP-Code 200 OK zurückgegeben. Im Body wird dann ein JSON COMRESULT-Objekt mit den Standardinformationen für Code 200 zurückgegeben.

```
HTTP/1.1 200 OK
Content-Length: 85
Content-Type: text/html
Date: Tue, 17 Mar 2015 23:53:42 GMT
Server: WebWare PoWeR Server
```

```
{ "COMRESULT":
  {
    "STATUS": 200,
    "CODE": "200 OK",
    "INFO": "CONNECTION CLOSED"
  }
}
```

## WWSVC-CURSOR Beenden

### /WWSVC/WWSERVICE/CURSORCLOSE/

Servicepunkt:

<https://MeineDomain.de/WWSVC/WWSERVICE/CURSORCLOSE/ServicePass/WWSVC-Cursor>

Wird ein WWSVC-CURSOR von einem Client nicht mehr benötigt so muss er diesen mit CURSORCLOSE freigeben. Dadurch werden von dem WWSVC-CURSOR belegte Ressourcen der WEBWARE freigegeben.

../WWSVC/WWSERVICE/CURSORCLOSE/123ServicePassID789/WWSVC-12345678-CURSOR

und nun eine komplette HTTP-Anfrage

```
GET /WWSVC/WWSERVICE/CURSORCLOSE/123SVP789/WWSVC-12345678-CURSOR
Host: myDomain.de
WWSVC-TS:
WWSVC-REQID:
WWSVC-APPID:
```

### Mögliche Rückgabe-Codes

- 406 Fehler kein gültiger ServicePass wurde übergeben
- 404 WWSVC Subsystem ist auf dem WW-Server nicht verfügbar bzw. ServicePass nicht bekannt, bzw. WWSVC-CURSOR nicht bekannt, siehe weitere Details in der Rückgabe
- 200 WWSVC-CURSOR wurde erfolgreich Beendet

### CLOSE OK

Der WW SVC Beendet den WWSVC-Cursor und gibt HTTP-Code 200 OK zurückgegeben. Im Body wird dann ein JSON COMRESULT-Objekt mit den Standardinformationen für Code 200 zurückgegeben.

```
HTTP/1.1 200 OK
Content-Length: 85
Content-Type: text/html
Date: Tue, 17 Mar 2015 23:53:42 GMT
Server: WebWare PoWeR Server
```

```
{ "COMRESULT":
  {
    "STATUS": 200,
    "CODE": "200 OK",
    "INFO": "WWSVC-CURSOR CLOSED"
  }
}
```

## **Ergebnis asynchroner Service-Funktion abrufen /WWSVC/WWSERVICE/GETASYNCRESLT**

../WWSVC/WWSERVICE/GETASYNCRESLT/123...ServicePassID...789/Async-Result-Handle  
Aufruf von GETASYNCRESLT mit einer gültigen Service-Pass-ID sowie gültigem Async-Result-Handle.

Wurde eine Service-Funktion zur Asynchronen Verarbeitung an den WEBWARE-Server (WWSVC) übergeben, so erhält man dabei einen Asynchronen-Result-Handle.

Der Asynchrone-Result-Handle wird mit Hilfe eines Header-Tags:

WWSVC-ASYNCHRON-HANDLE: WWSVC-ASYNC-06DA94C8

sowie in der Service-Pass Struktur wird dabei auch eine Variable

"WWSVC\_ASYNC\_HRON\_HANDLE": WWSVC-ASYNC-06DA94C8

Mit diesem Asynchron-Handle kann man dann beim WEBWARE-Server zeitlich versetzt das Ergebnis der Abfrage abholen.

Dabei kann es zu folgenden Zuständen kommen:

- Bearbeitung ist abgeschlossen, Ergebnis steht bereit (HTTP - Code der Verarbeitung)
- Noch in Bearbeitung (HTTP-Code 202)
- Async-Result-Handle ist ungültig (HTTP-Code 400)

### **Mögliche Rückgabe-Codes**

- |     |                                                                                    |
|-----|------------------------------------------------------------------------------------|
| 400 | Funktion nicht verarbeitbar/nicht bekannt, bzw. Async-Result-handle nicht gültig   |
| 404 | WWSVC Subsystem ist auf dem WW-Server nicht verfügbar                              |
| 202 | Asynchron Auftrag wurde eingelastet und wir erhalten einen Asynchron-Handle        |
| 200 | Ausführung erfolgreich, Ergebnis wird übergeben Asynchrone Operation abgeschlossen |

Beispiel einer Anfrage die Asynchron bearbeitet werden soll:

```
PUT /WWSVC/EXECJSON HTTP/1.1
Content-Length: 456
WWSVC-EXECUTE-MODE: ASYNCHRON
WWSVC-REQID: 2
Content-Type: application/json
WWSVC-HASH: 0832f01df9b4e00d4820d1618017dac0
WWSVC-TS: Wed, 05 Aug 2015 08:26:42 GMT
WWSVC-SESSION-TOKEN: 0eaa1c78947cb822006bf0b9791eb028

{  "WWSVC_PASSINFO":
  {    "SERVICEPASS":"45c92b5959a3dd760ced41a7db5748b5",
      "APPHASH":"0832f01df9b4e00d4820d1618017dac0",
      "TIMESTAMP":"Wed, 05 Aug 2015 08:26:42 GMT",
      "REQUESTID":2,
      ""EXECUTE_MODE":"ASYNCHRON""
    },
  "WWSVC_FUNCTION":
  {    "FUNCTIONNAME":"GET_ARTIKEL",
      "PARAMETER": ...
  }
}
```

Beispiel einer Antwort auf eine Asynchrone Anforderung:

Dabei wird der Asynchron-Result-Handle als Header-Feld sowie in der COMRESULT-Struktur übergeben

```
HTTP/1.1 202 Accepted
Content-Length: 158
Content-Type: application/json
Date: Wed, 05 Aug 2015 10:27:54 GMT
WWSVC-ASYNCHRON-HANDLE: WWSVC-ASYNC-045A9220

{ "COMRESULT":
  {
    "STATUS": 202,
    "CODE": "ASYNCHRON-FUNCTION-ACCEPTED",
    "INFO": "ASYNCHRON FUNCTION ACCEPTED",
    "WWSVC_ASYNCHRON_HANDLE": WWSVC-ASYNC-045A9220
  }
}
```

Beispiel: Abrufen des Ergebnisses einer Asynchron in Auftrag gegebenen Service-Funktion

Hierzu wird die hier beschriebene WWSVC-Service-Funktion

./WWSVC/WWSERVICE/GETASYNCRESULT/Pass-ID/Asynch-Result-Handle

verwendet. Bei dieser muss dabei ein gültiger Service-Pass sowie ein gültiger Asynchron-Result-Handle übergeben werden.

```
GET /WWSVC/WWSERVICE/GETASYNCRESULT/45c92b5959a3dd760ced41a7db5748b5/WWSVC-ASYNC-045C16D0 HTTP/1.1
WWSVC-SESSION-TOKEN: 0eaa1c78947cb822006bf0b9791eb028
WWSVC-EXECUTE-MODE: SYNCHRON
WWSVC-REQID: 5
WWSVC-HASH: a5e2d674244363925deec710ad43070b
WWSVC-TS: Wed, 05 Aug 2015 08:28:53 GMT
```

Je nach Bearbeitungszustand der asynchron in Auftrag gegebenen Service-Funktion kann es hier nun folgende Ergebnisse geben:

**Bearbeitung ist abgeschlossen**

Bei Zugriff auf eine fertig bearbeitete Service-Funktion erhalten Sie das Ergebnis wie gewohnt zurück. Durch den einmaligen erfolgreichen Zugriff auf das Ergebnis wird der Asynchron-Result-Handle ungültig und weitere Abrufe mit diesem sind nicht mehr möglich.

HTTP/1.1 200 OK

Content-Length: 120

Content-Type: application/json

Date: Wed, 05 Aug 2015 10:28:08 GMT

```
{"COMRESULT": {"STATUS": 200, "CODE": "200 OK", "INFO": "OK"},
```

```
  "ARTIKELLISTE":  
  {  
    "ANZAHL": "0"  
  }  
}
```

Der HTTP-Rückgabestatus der asynchronen Service-Funktion wird als Ergebnis gesetzt.

**Bearbeitung ist noch nicht abgeschlossen (HTTP-Code 202)**

Ist die Bearbeitung der Service-Funktion noch nicht abgeschlossen, so erhalten Sie den Status-Code 202 mit weiteren Informationen in der COMRESULT-Struktur

HTTP/1.1 202 Accepted  
Content-Length: 207  
Content-Type: application/json

```
{
  "COMRESULT":
  {
    "STATUS": 202,
    "CODE": "202 Accepted",
    "INFO": "ASYNCHRON-SVF-IN-PROGRESS",
    "ERRORCODE": 202,
    "ERRORLINK": "DOCWWSVC/INFO.HTML/#00202",
    "ERRORINFO": "ASYNCHRON Service Function in Progress"
  }
}
```

Sie haben damit die Möglichkeit zu einem späteren Zeitpunkt erneut mit dem Asynchron-Result-Handle zuzugreifen um das Ergebnis abzurufen.

**Zugriff fehlgeschlagen (HTTP-Code 400)**

Ist der Asynchron-Result-Handle nicht gültig, so erhalten Sie einen HTTP-Fehlercode 400 mit weiteren Informationen in der COMRESULT-Struktur. Dabei wird der zusätzliche Fehlercode (ERRORCODE:) 50500 angezeigt.

HTTP/1.1 400 Bad Request  
Content-Length: 204  
Content-Type: application/json

```
{
  "COMRESULT":
  {
    "STATUS": 400,
    "CODE": "400 Bad Request",
    "INFO": "ASYNCHRON-HANDLE-NOT-KNOWN",
    "ERRORCODE": 50500,
    "ERRORLINK": "DOCWWSVC/ERR.HTML/#50500",
    "ERRORINFO": "ASYNCHRON HANDLE IST NOT KNOWN"
  }
}
```

## Verbindungsrouting von Browser zu Fremdsystem /WWSVC/WWSERVICE/WEBPROXY

../WWSVC/WWSERVICE/WEBPROXY/Ziel-URL

Ist ein Zugriff aus dem Browser durch CORS nicht möglich, so bietet der WEBWARE-Server die Möglichkeit der Einrichtung eines HTTPS-Proxy's mit dem einzelne Verbindungen zu Fremdsystemen aufgeführt werden können.

Wenn zum Beispiel Ihre Anwendung vom WEBWARE Server ausgeliefert wird und Sie versuchen aus dem Browser auf andere Domain's, so wird das im Browser unter Umständen unterbunden. Dies liegt dann daran das die "andere" Domain also der andere WEB-Server keine Erlaubnis zum Verbindungsaufbau gibt wenn Ihre Anwendung nicht von diesem anderen WEB-Server geladen wurde.

Nähere Hinweise zu CORS finden Sie "Kurze Einführung in CORS".

Um den WEBWARE WWSVC-WEB-Proxy zu aktivieren muss neben WWSVC auch der System-Wert

WW Service WEB Proxy erlauben

1

im Bereich Konfiguration>SystemKonfiguration>WWSVC WEBWARE Services ..>aktiviert werden.

Der WWSVC WEBPROXY hat die Funktion die Daten von Ihrer Browser Anwendung zu Empfangen. Dabei wird beim Aufruf, also in der URL die gewünschte ZIEL-URL in einem speziellen Format übergeben. Das Format ist notwendig damit das HTTP-Protokoll nicht gestört wird.

Neben der ZIEL-URL wird der Inhalt (Body) komplett sowie einige Header-Felder aus der Anfrage übernommen.

### Beispiel eines Zugriffs auf Afterbuy:

Hierbei wird die Interface-Adresse

<https://api.afterbuy.de/afterbuy/ABInterface.aspx>

in

<https%5E%60%60api.afterbuy.de%60afterbuy%60ABInterface.aspx>

umgesetzt. Dabei werden die Doppelpunkte ":" mit der Kennung %5E sowie die Verzeichnistrenner "/" mit der Kennung %60 ersetzt.

Werden Daten vom entfernten System empfangen so werden diese mit 2 Änderungen direkt an Ihr Programm übergeben. Etwa vorhanden X-Frame-Options werden überschrieben, sowie es wird eine "Access-Control-Allow-Origin: \*" Regel eingefügt.

### Mögliche Rückgabe-Codes

400 Wenn die Übergabeparameter Fehlerhaft sind, (Kein HTTPS, usw.) bzw. keine Ausführung

404 Wenn WEBPROXY nicht erlaubt ist, bitte System-Wert überprüfen.

Sonstige Fehler- / Statuscodes kommen dann vom entfernten System



## Beispiel Afterbuy Abruf

### ***Abruf einer Afterbuy-Schnittstellen-XML:***

```
POST /WWSVC/WWSERVICE/WEBPROXY/https%5E%60%60api.afterbuy.de%60afterbuy%60ABInterface.aspx HTTP/1.1
Host: meine-webware.de
Connection: keep-alive
Content-Length: 395
Pragma: no-cache
Cache-Control: no-cache
Origin: https://meine-webware.de
Content-Type: application/xml
Accept: */*
DNT: 1
Referer: https://meine-webware.de /wwapp-test/index.html
Accept-Encoding: gzip, deflate, br
Accept-Language: de-DE,de;q=0.8,en-US;q=0.6,en;q=0.4

<?xml v.....
```

### ***Aus dieser Anfrage wird dann folgender HTTPS Aufruf erzeugt und entsprechend abgesetzt:***

```
POST /afterbuy/ABInterface.aspx HTTP/1.1
Content-Length: 395
Content-Type: application/xml
Host: api.afterbuy.de
Referer: https://api.afterbuy.de/afterbuy/ABInterface.aspx
Pragma: no-cache
Cache-Control: no-cache

<?xml v.....
```

### ***Empfangen der Antwort und Rückgabe an Ihre Anwendung***

Die empfangenen Daten des entfernten Rechners werden dann mit 2 Header-Anpassungen als Antwort an Ihre Anwendung zurückgegeben..

```
HTTP/1.1 200 OK
Cache-Control: private
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
Strict-Transport-Security: max-age=31536000; includeSubDomains
Set-Cookie: ASP.NET_SessionId=wvj0fkdjgt3otukzkzpxasw; path=/; HttpOnly
X-Powered-By: ASP.NET
Access-Control-Allow-Origin: *
Date: Tue, 07 Jun 2016 07:16:30 GMT

6cb4
<?xml version="1.0" encoding="utf-8"?>
<Afterbuy>
  <CallStatus>Success</CallStatus>
```

## WWSVC/EXEC Funktion

Mit der WWSVC EXEC Funktion kann nun für ein ServicePass eine zugehörige Funktion ausgeführt werden. Dabei können nur Funktionen ausgeführt werden die in der Funktionsgruppe für die SecuredApp definiert sind.

Es sind je nach Aufrufformat 2 Zugriffsfunktionen definiert. /EXECURL und /EXECJSON

/EXECURL	Aufruf mit Funktionsbeschreibung und Parametern in der URL
/EXECJSON	Aufruf mit Funktionsbeschreibung und Parametern im Body als JSON

## HTTP / REST Funktionen

Um Funktionen auszuführen haben Sie die Möglichkeit diese per normalen PUT/POST auszulösen. Dabei ist zu Beachten das der WWSVC-Server eingehende Funktionsnamen bei Zugriff auf eine Datenobjekt wie zum Beispiel ARTIKEL, BELEGE, (...) oder WORKFLOW je nach HTTP-Befehl auf festgelegte Funktionen routet.

Wird bei einem Funktionsaufruf direkt eine Funktion für ein Datenobjekt angegeben so entfällt das weitere Routen.

Beispiel (zur Vereinfachung im EXECURL Format):

Wenn Sie eine Funktion ausführen auf das Datenobjekt ARTIKEL und geben dabei keine weitere Ausführungs-Funktion an wie zum Beispiel (ARTIKEL.UPDATE, ARTIKEL.INSERT, ARTIKEL.DELETE ARTIKEL.GET), dann wird anhand des HTTP-Befehls die Ziel-Funktion automatisch angefügt.

PUT	/WWSVC/SVP/ARTIKEL	ARTIKEL.UPDATE
POST	/WWSVC/SVP/ARTIKEL	ARTIKEL.INSERT
GET	/WWSVC/SVP/ARTIKEL	ARTIKEL.GET
INSERT	/WWSVC/SVP/ARTIKEL	ARTIKEL.INSERT
DELETE	/WWSVC/SVP/ARTIKEL	ARTIKEL.DELETE

Beachten Sie das bei Ausführung mit dem GET-Befehl teilweise keine Übergabe von JSON-Parametern aus Browsern erfolgt.

Sie können also für den Zugriff auf die /WWSVC/EXEC... Funktionen folgende HTTP-Befehle verwenden:

- PUT entspricht Datenobjekt . UPDATE
- POST entspricht Datenobjekt . INSERT
- GET entspricht Datenobjekt . GET
- INSERT entspricht Datenobjekt . INSERT
- DELETE entspricht Datenobjekt . DELETE
- UPDATE entspricht Datenobjekt . UPDATE
- EXEC entspricht Datenobjekt . EXEC (Ausführen von Workflows usw.)
- OPTIONS Abfragen der API-JSON Struktur für das angegebene Objekt

## Hinweis bei Verwendung von Benutzer-Listen

Ist der Zugriff für eine Service-Anwendung per Benutzer/Passwort Liste geschützt und ist die Anmeldung bei Zugriff aktiviert, so muss vor dem ersten Zugriff ein Connect mit Übergabe von Benutzer/Passwort erfolgen, damit die Verbindung über einen WWSVC-SESSION-TOKEN Cookie geschützt werden kann.

Hier ist ebenfalls zu Beachten das dieser WWSVC-SESSION-TOKEN nach Vorgabe ablaufen kann und bei Zugriff mit einem abgelaufenen WWSVC-SESSION-TOKEN eine "401-Authorization Required" Fehlermeldung zurückgegeben wird.

**Hinweis für die Bearbeitungsart (Synchron/Asynchron)**

In der Basiskonfiguration der WEBWARE Services werden alle Anfragen SYNCHRON ausgeführt. Dabei wird die Verbindung beim Start einer Service-Funktion solange beim WEBWARE-Server gehalten bis ein Ergebnis für die Service-Funktion vorliegt.

Um Ihnen und damit Ihren Anwendungen bei der Verwendung von Service-Funktionen mehr Flexibilität zu verschaffen, ist es auch möglich Service-Funktionen Asynchron abzusetzen, wobei dann die Verbindung direkt nach der erfolgreichen Einlastung der Service-Funktion mit einem zugehörigen Asynchron-Handle zurückgegeben wird. Es ist dann möglich zu einem späteren Zeitpunkt mit Hilfe des Asynchron-Handle das Ergebnis der Service-Funktion abzurufen. Desweiteren steht ein weiterer Modus zur Verfügung mit dem kein Abrufen des Asynchronen Ergebnisses vorgesehen ist.

Der Ausführungs-Modus für eine Service-Funktion kann entweder über ein HTTP-Header Feld

WWSVC-EXECUTE-MODE: ASYNCHRON,

bzw. in der JSON-Struktur für den Service-Pass die Variable

"EXECUTE\_MODE":"ASYNCHRON"

gesetzt werden.

Folgende Werte sind hier möglich:

- SYNCHRON (Standard-Wert, bzw. wenn Wert nicht angegeben, bzw. wenn nicht erlaubt)
- ASYNCHRON Asynchrone-Verarbeitung anfordern
- ASYNCHRON\_NO\_RESULT Asynchrone-Verarbeitung ohne Möglichkeit Ergebniszugriffs

**Hinweis für die Verwendung von WWSVC-Cursor**

WWSVC erlaubt die Verwendung von WWSVC-Cursor. Dabei ist es möglich das Ergebnis eines Funktionsaufrufs "Seitenweise" abzurufen, also kann aus einem Abfrage-Ergebnis nur ein Teil abgerufen werden.

Wichtig ist hier das Sie immer die maximal gewünschte Anzahl Sätze für das Ergebnis mit der Header-Variable WWSVC-ACCEPT-RESULT-MAX-LINES angeben.

Sie können mit dem HTTP-Header WWSVC-CURSOR bzw. mit der Variable GET\_WWSVC\_CURSOR in der Service-Pass JSON-Struktur die WWSVC-Cursor Steuerung durchführen. (In der Folge werde ich nur den HTTP-Header WWSVC-CURSOR verwenden)

Es gibt dabei 3 mögliche Zustände die der WWSVC-CUROSr annehmen kann.

- WWSVC-CURSOR=CREATE
- WWSVC-CURSOR=[WWSVC-ABCDEFGH-CURSOR]
- WWSVC-CURSOR=Leer bzw. nicht vorhanden.

Wenn Sie einen Cursor erzeugen wollen, so wird dies mit folgender Angabe veranlasst:

WWSVC-CURSOR=CREATE

WWSVC-ACCEPT-RESULT-MAX-LINES=30

Werden wie hier im Beispiel mehr als 30 Ergebnis Sätze gefunden, so wird ein Cursor-Zugriff erzeugt und im Service-Punkt vorgehalten. Sie erhalten dann neben der ersten Ergebnissätze in der Rückgabe den Cursor-Zugriff in einem Cookie übergeben

WWSVC-CURSOR=[Kennung für Cursor, Installations-Abhängig.]

Wollen Sie die nächsten Sätze für einen offenen Cursor lesen, so erreichen Sie das mit der Angabe des Cursor-Zugriffs, sowie der gewünschten Ergebnis-Anzahl

WWSVC-CURSOR=[Kennung für Cursor, Installations-Abhängig.]

WWSVC-ACCEPT-RESULT-MAX-LINES=30

Werden hier weniger als die gewünschten Sätze gefunden, so wird der Cursor automatisch geschlossen, und Sie erhalten die Kennung **WWSVC-CURSOR=CLOSED** mit dem Ergebnis zurück.

Hinweis: Es ist zur Laufzeit immer nur ein Zugriff auf einen Cursor erlaubt, ist bereits für einen Cursor ein Service-Anfrage im System und noch nicht fertig bearbeitet, so wird ein erneuter Zugriff auf diesen Cursor mit einem Fehler quittiert.

Ein Cursor kann auf 5 Arten geschlossen werden:

- WWSVC-CURSOR=CLOSE-[Kennung für Cursor, Installations-Abhängig.]
- /WWSVC/CURSORCLOSE/[Service-Pass]/Cursor
- Automatisch durch Erreichen des Ende des Ergebnis (Rückgabe WWSVC-CURSOR=CLOSED)
- Überschreiten der maximalen Leerlauf / Wartezeit für einen WWSVC-CURSOR
- Programm Interner Absturz / Neustart des Prozess-Knotens

## WWSVC/EXECURL Funktion

Mit diesem Funktionsaufruf können Sie direkt den Funktionsnamen sowie die Funktionsparameter in der HTTP-URL verwenden. Dieser Aufruf benötigt kein Body im HTTP-Call.

### Beispiel der URL-Parameter für einen Funktionsaufruf:

Aufbau der ServicePunktes:

WWSVC            Service-Punkt

EXECURL        Ausführen von Funktion mit URL-String

PASS-ID         ServicePassID für die SecuredApp

Funktionsname   Funktion die ausgeführt werden soll, muss Bestandteil der APP-Funktiongruppe sein

Parameter-Liste   Je nach Funktion werden hier die Aufrufparameter an die Funktion angegeben

## Hinweis zur Verwendung und Angabe von Parametern:

### Position/Parameter-Name im Aufruf

Um eine Zuordnung des Parameters zu dem Funktionsparameter der aufgerufenen Funktion zu machen, haben Sie 2 Möglichkeiten.

- Angabe eines Parameter-Namens sowie ein Ist-Gleich-Zeichen und Parameter-Wert `PARAMETER=WERT`
- Direkt den Parameter-Wert `../SET_ARTIKEL_LAGER/1000001/"Buchen"/`

Im ersten Fall, also Angabe von `Name=Wert`, wird direkt der Funktionsparameter unabhängig von seiner Position im URL-Aufruf gesetzt. Also ist zum Beispiel der Parameter "PARAMETER4" der 4. Parameter so wird dieser bei Aufruf: `../STOREIMAGE/PARAMETER4="TEST"/DATEINAME=bildklein.jpg/` korrekt als 4. Parameter für die Funktion zugeordnet.

Im zweiten Fall, also keine Angabe eines Parameter-Namens, wird der enthaltene Wert direkt anhand seiner Position als Funktionsparameter zugeordnet.

### Angabe von Texten und dem URL-Trenner und Gleichheitszeichen

Der URL-Aufrufstring wird mit Hilfe des URL-Trenners / unterteilt. Daher ist es nicht möglich in einen normalen Parameter diesen Trenner als Wert aufzunehmen. Ebenso ist es nicht möglich das Gleichheitszeichen das zur Zuweisung eines Wertes zu einem Funktionsparameter Namens dient anzugeben.

Beispiel Pfad-Angabe für Datei `/BILDNAME=bilder/meinbild.png/DELETE`

Beispiel Druck-Text `/ADDDPRINTTEXT/100/100/Gesamtwert der Dienstleistung = 423,00 EUR/`

In diesem Fall ist es notwendig den Parameter-Wert in Anführungszeichen zu setzen um damit Texte die die Sonderzeichen / und = enthalten korrekt verarbeiten zu können.

Korrekte Verwendung:

Beispiel Pfad-Angabe für Datei `/BILDNAME="bilder/meinbild.png"/DELETE`

Beispiel Druck-Text `/ADDDPRINTTEXT/100/100/"Gesamtwert der Dienstleistung = 423,00 EUR"/`

**Aufruf /WWSVC/EXECURL**

Die Aufrufparameter werden hierbei wie oben beschrieben übergeben.

```
GET /WWSVC/EXECURL/63daf57370a3faf9e71f4260155d01a9/TESTFUNC/PAR1=Value/PAR2="Val/t-12" HTTP/1.1
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
WWSVC-HASH: 49b7b75dac82a56d95889a5e9b76eb78
WWSVC-REQID: 46
WWSVC-TS: Wed, 18 Mar 2015 00:00:11 GMT
```

**Antwort auf /WWSVC/EXECURL**

Der WW-Server antwortet mit HTTP Statuscode 200 wenn die Funktion erfolgreich ausgeführt werden konnte.

```
HTTP/1.1 200 OK
Content-Length: 350
Content-Type: text/html
Date: Wed, 18 Mar 2015 00:00:11 GMT
Server: WebWare PoWeR Server
```

```
{ "COMRESULT":
  {
    "STATUS": 200,
    "CODE": "200 OK",
    "INFO": "OK TESTFUNCTION, get Color-Values"
  },
  { "COLORVALUES":
    { color: "red", value: "#f00"},
    { color: "green", value: "#0f0"},
    { color: "blue", value: "#00f"},
    { color: "cyan", value: "#0ff"},
    { color: "magenta", value: "#f0f"},
    { color: "yellow", value: "#ff0"},
    { color: "black", value: "#000"}
  }
}
```

Falls die Funktion nicht ausgeführt werden kann, so gibt es in der COMRESULT-Struktur der Antwort jeweils die Info warum die Funktion nicht ausgeführt werden konnte.

```
HTTP/1.1 404 Resource not found
Content-Length: 116
Content-Type: text/html
Date: Wed, 18 Mar 2015 00:07:20 GMT
Server: WebWare PoWeR Server
```

```
{ "COMRESULT":
  {
    "STATUS": 404,
    "CODE": "404 Resource not found",
    "INFO": "ERROR SERVICEPASS IS NOT ALLOWED TO RUN"
  }
}
```

## WWSVC/EXECJSON Funktion

Mit der WWSVC/EXECJSON können Sie ebenfalls einen Funktionsaufruf machen mit der Besonderheit das alle Cookies und Parameter als JSON-Objekt übergeben werden. Der Aufruf sollte in dem Fall als HTTP PUT Befehl ausgeführt werden. Der WW-Server erlaubt jedoch auch eine GET-Funktion mit Datenübergabe im Body des HTTP-Request.

### WWSVCEXEC JSON-Objekt für Funktionsaufruf

Folgende Parameter werden im JSON-Objekt erwartet. Alle Parameter werden in Großschreibung angegeben.

#### WWSVC\_PASSINFO

SERVICEPASS	ServicePass für diesen Aufruf
APPHASH	Berechneter Hash-Wert mit App-Secret und Zeitstempel
TIMESTAMP	Zeitstempel
REQUESTID	fortlaufende Zugriffsnummer der Verbindung
* Optionale Parameter weiter unten beschrieben, falls in JSON angegeben, hat dieses Vorrang	
[GET_RESULT_MAX_LINES	Ergebnis einschränken / Bei 0 wird WW-Systemwert gesetzt]
[GET_WWSVC_CURSOR	Cursor in Ergebnis]
[GET_RESULT_TYPE	Form der Rückgabedaten]
[APPID	Name der Client-Anwendung zur Information/Debugging]
[SESSION_TOKEN	Session Token für Zugriffssteuerung]
[EXECUTE_MODE]	Wie soll die Ausführung erfolgen (DEFAULT: SYNCHRON)
[WWSVC_DYNOB]	Verwendung von dynamischen Objekten]
[..] <i>Optionale Parameter weiter unten beschrieben, falls in JSON angegeben, hat dieses Vorrang</i>	

#### WWSVC\_FUNCTION

FUNCTIONNAME	Funktionsname der ausgeführt werden soll
REVISION	[Optional] Verwende Revision, 0=Default
PARAMETER	Array of Functionparameter
POSITION	[Optional] Nummer des Parameter für Übergabe
PNAME	[Optional] Name für Parameter
PTYPE	[Optional] Type des Datentypes (S=String,N=Number)
PCONTENT	Inhalt des Parameter

Beispiel einer JSON-Anfrage

PUT /WWSVC/EXECJSON HTTP/1.1

Connection: keep-alive

Content-Length: 454

Content-Type: application/json

```
{
  "WWSVC_PASSINFO":
  {
    "SERVICEPASS": "0778e54d57f23b58bb0cdb2cb1e45691",
    "APPHASH ": "88e49c05f9c49d86a2465117ec7e9de1",
    "TIMESTAMP": "Tue, 17 Mar 2015 23:33:57 GMT",
    "REQUESTID": "42",
    "SESSION_TOKEN": "88e49c05f9c49d86a2465117ec7e9de1"
  },
  "WWSVC_FUNCTION":
  {
    "FUNCTIONNAME": "HELLO_WORLD",
    "PARAMETER":
    [
      {"POSITION": "1", "PNAME": "USERNAME", "PCONTENT": "Aexel Bachli"},
      {"POSITION": "2", "PNAME": "LANGAUGE", "PCONTENT": "DEUTSCH"}
    ]
  }
}
```

### Mögliche Rückgabe-Codes

- |     |                                                                                       |
|-----|---------------------------------------------------------------------------------------|
| 400 | Funktion nicht verarbeitbar/nicht bekannt, Nähere Hinweise in der COMRESULT           |
| 406 | Fehler kein gültiger ServicePass wurde übergeben                                      |
| 404 | WWSCVC Subsystem ist auf dem WW-Server nicht verfügbar bzw. ServicePass nicht bekannt |
| 200 | Ausführung erfolgreich                                                                |



## Cookies und Transport Sicherheit

Zwischen Client und Server wird bei Registrierung ein öffentlicher Schlüssel (PASS-ID) sowie ein geheimer Schlüssel (APP-Secret) ausgetauscht.

Der PASS-ID Schlüssel dient zur öffentlich Übergabe des Service-Passes der verwendet werden soll. Dadurch ist es möglich in einer Client-Anwendung mehrere Service-Pässe zu verwenden. Der Service-Pass wird bei EXEC Funktion immer als Teil der Aufrufparameter übergeben.

Der APP-Secret Schlüssel bleibt auf Seiten des Client's und dient als Basis zu Erstellung eines MD5-Hashwertes.

In der aktuellen Version wird die Validierung der Nachricht durch einen einfache HASH-Wert-Erstellung sichergestellt. Dabei wird jede neue Nachricht mit einem neu berechneten HASH-Wert übergeben. Dieser berechnet sich aktuell aus dem APP-Secret + aktueller Zeitstempel der als Cookie WWSVC-TS übergeben wird.

Beispiel für notwendige WWSVC-Cookies die bei jeder HTTP-Anfrage (bis auf die Registrierung) mit angegeben werden müssen

WWSVC-HASH: 49b7b75dac82a56d95889a5e9b76eb78

WWSVC-REQID: 46

WWSVC-TS: Wed, 18 Mar 2015 00:00:11 GMT

Optional

WWSVC-APPID: "mobile Auftragserfassung"

WWSVC-ACCEPT-RESULT-TYPE: "JSON"

WWSVC-ACCEPT-RESULT-MAX-LINES:100

### WWSVC-REQID

### laufende Request-ID

Hier wird eine fortlaufende Nummer der Anfragen übergeben. Diese kann später auch vom WW-Server für Validierung durch Sequenz Überprüfung verwendet werden.

### WWSVC-TS

### Aktueller Zeitstempel

Hier wird der aktuelle Zeitstempel übergeben der zum Zeitpunkt des Senden beim Client erstellt wurde. Dieser Zeitstempel wird zur Zeit verwendet um mit dem APP-Secret einen MD5-Hash zu Berechnen.

### WWSVC-HASH

### Meldungs Hash für Validierung

Mit diesem 32-Byte MD5-Hash-Wert wird auf Seiten des WW-Servers geprüft ob diese Meldung zu dem Service-Pass gehört und ob diese gültig ist.

Die Berechnung erfolgt mit MD5(App-Secret+WWSVC-TS)

**WWSVC-APPID****Name für Client-Anwendung**

Hier kann der Client einen Namen für seine Anwendung vorgeben.

Beispiel: WWSVC-APPID: "SE:mobiles eBanking"

**WWSVC-PASSID****Service-Pass optional HTTP-Header**

Hier kann der Client bei Aufruf von EXECJSON auch den Service-Pass als Header-Information setzen, um die Funktionsaufrufparameter nicht erneut erstellen zu müssen

Beispiel: WWSVC-PASSID: "49b7b75dac82a56d95889a5e9b76eb78"

**WWSVC-ACCEPT-RESULT-TYPE****Form der Rückgabedaten**

Hier kann man angeben, in welchem Format die Rückgabedaten gewünscht sind. Mögliche Formate sind:

- JSON JSON-Struktur ohne Leere Zeilen/EndeLeerzeichen (DEFAULT)
- JSON-FILLED JSON-Struktur ohne Leer Zeilen, Felder mit Leerzeichen aufgefüllt
- JSON-DETAIL JSON-Struktur mit Leeren Zeilen und Felder aufgefüllt
- XML Es wird eine XML-Struktur zurückgegeben
- BIN \* Rückgabe von Bildern/PDF usw. erwartet.
- DAT \* Noch nicht definiert ..
- CSV \* Noch nicht definiert ..

Beispiel: WWSRV-ACCEPT-RESULT-TYPE="JSON"

**WWSVC-ACCEPT-RESULT-MAX-LINES****Ergebnis einschränken**

Wenn die Anzahl Ergebniszeilen nicht abzuschätzen sind, so kann man mit diesem Cookie die Anzahl des Ergebnisses beschränken. Standardmäßig geben alle Service-Funktionen alle gefundenen Sätze die zu der Anforderung passen zurück, [WWSVC-ACCEPT-RESULT-MAX-LINES=0 oder nicht vorhanden].

Da dies jedoch bei sehr großen Datenbeständen zu Laufzeitproblemen führen kann, (Bspl Abruf von allen Artikeln (500000..)) gibt es im System-Cockpit einen System-Wert mit dem bei keiner Vorgabe die maximale Anzahl der Sätze gesetzt werden.

Daten	
Beschreibung	Systemwert
Setze RESULT_MAX_LINES bei 0 auf	100

Will man nur eine Bestimmte Anzahl von Ergebnis-Sätzen so kann man dies entsprechend mit WWSVC-ACCEPT-RESULT-MAX-LINES=123 (maximal 123 Stück, ähnlich wie bei SQL-Datenbank Top 123) eingrenzen

Beispiel: Gebe mir alle Artikel aber, Maximal 123 Stück ..

"WWSVC-ACCEPT-RESULT-MAX-LINES": 123

## WWSVC-CURSOR

## Cursorfunktionen

Mit dieser Vorgabe kann aus einem Ergebnis nur ein Teil abgerufen werden. Damit ist es zum Beispiel möglich ein sehr großes Result-Set in einer Tabelle darzustellen.

Sie können mit dem Befehl WWSVC-CURSOR solche großen Result-Set's verwalten. Wichtig ist hierbei die Angabe der Anzahl die maximal pro Zugriff zurückgegeben werden sollen (WWSVC-ACCEPT-RESULT-MAX-LINES=xx).

Wenn Sie einen Cursor erzeugen wollen, so wird dies mit folgender Angabe veranlasst:

```
WWSVC-CURSOR=CREATE  
WWSVC-ACCEPT-RESULT-MAX-LINES=30
```

Werden wie hier im Beispiel mehr als 30 Ergebnis Sätze gefunden, so wird ein Cursor-Zugriff erzeugt und im Service-Punkt vorgehalten. Sie erhalten dann neben der ersten Ergebnissätze in der Rückgabe den Cursor-Zugriff in einem Cookie übergeben

WWSVC-CURSOR=[Kennung für Cursor, Installations-Abhängig.]

Wollen Sie die nächsten Sätze für einen offenen Cursor lesen, so erreichen Sie das mit der Angabe des Cursor-Zugriffs, sowie der gewünschten Ergebnis-Anzahl

```
WWSVC-CURSOR=[Kennung für Cursor, Installations-Abhängig.]  
WWSVC-ACCEPT-RESULT-MAX-LINES=30
```

Werden hier weniger als die gewünschten Sätze gefunden, so wird der Cursor automatisch geschlossen, und Sie erhalten die Kennung **WWSVC-CURSOR=CLOSED** mit dem Ergebnis zurück.

Hinweis: Es ist zur Laufzeit immer nur ein Zugriff auf einen Cursor erlaubt, ist bereits für einen Cursor ein Service-Anfrage im System und noch nicht fertig bearbeitet, so wird ein erneuter Zugriff auf diesen Cursor mit einem Fehler quittiert.

## WWSVC-DYNOBJ

## Verwendung Dynamische Objekte

Bei Verwendung von MEMTAB's ist es möglich dynamische Objekte zu Erstellen die über die Dauer des aktuellen WWSVC-Zugriffs Gültigkeit haben. Diese Objekte sind dann zur Laufzeit an einen bestimmten Prozess-Knoten gebunden. Unter Angabe des WWSVC-DYNOBJ Header Feldes ist es möglich diese bestehenden dynamischen Objekte erneut anzusprechen, dies ist ebenso mit der Variable WWSVC\_DYNOBJ in der WWSVC\_PASSINFO JSON Struktur möglich.

Beispiel: Sie erstellen eine MEMTAB aller Artikel mit Lagerbestand 0 mit dem Namen ARTIKELTAB. Diese MEMTAB erzeugt dann eine WWSVC-DYNOBJ Kennung. In der Folge können nun weitere Service-Anfragen mit Hilfe der WWSVC-DYNOBJ Kennung auf diese MEMTAB ausgeführt werden.

MEMTAB.ARTIKELTAB.GET(..),MEMTAB.ARTIKELTAB.PRINT(..) usw..

Da diese MEMTAB nur in einem Prozess-Knoten bereit steht werden mit der WWSVC-DYNOBJ Kennung die Zugriffe zu diesem Prozess-Knoten geroutet. Dabei ist es möglich in der Folge mehrere Dynamische Objekte zu erzeugen, welche dann alle auf diesem Prozess-Knoten angelegt werden.

Eine Verwendung Dynamischer Objekte ist zur Laufzeit genau für einen Service-Pass gültig und je Service-Pass kann zu einem Zeitpunkt nur eine Verwendung Dynamischer Objekte aktiv sein. Dies ist per Systemwert an/abschaltbar, so dass Aktuell im Standard auch mehrere Dynamische Objekte je Service-Pass gleichzeitig aktiv sein können.

Die Verwendung Dynamischer Objekte ist so lange gültig wie bestehende Dynamische Objekte vorhanden sind. Wird zum Beispiel die letzte MEMTAB für ein Dynamisches Objekt gelöscht, so wird auch der Zugriff WWSVC-DYNOBJ ungültig und gelöscht.

Ebenso gibt es im System-Cockpit einen Time-Out Wert mit dem die maximale Lebensdauer eines inaktiven Dynamischen Objektes vorgegeben werden kann.

Es ist möglich bei der Verwendung von Dynamischen Objekten auch WWSVC-CURSOR zu Verwenden. Hier ist zu Beachten das die Dynamischen Objekte eine höhere Priorität haben und der WWSVC-CURSOR nach dem Dynamischen Objekt erstellt werden muss. Ist der WWSVC-CURSOR älter so sollte auf die Übergabe der WWSVC-DYNOBJ Vorgabe verzichtet werden, damit der Cursor ohne Fehler angesprochen werden kann.

### **WWSVC-SESSION-TOKEN**

Wird für den Zugriff auf eine SecuredApp vom Administrator festgelegt, das der Zugriff per Benutzer-Liste und nur mit einem gültigen Session-Token erfolgen darf, so muss der Client diesen Session-Token mit dem Befehl CONNECT anfordern. (Siehe weiter oben)

/WWSVC/WWSERVICE/CONNECT

In der Folge muss dann bei jedem Zugriff dieser WWSVC-SESSION-TOKEN übergeben werden. Falls der WWSVC-SESSION-TOKEN ungültig wird (Zeitablauf, bzw. Zugriff des gleichen Service-Pass ohne WWSVC-SESSION-TOKEN) erhält der Client die Fehlermeldung das der WWSVC-SESSION-TOKEN neu angefordert werden muss, mit dem Status-Code 401 AUTHORIZATION REQUIRED

**WWSVC-EXECUTE-MODE**

Mit dem WWSVC-EXECUTE-MODE kann angegeben werden ob eine Service-Funktion SYNCHRON (Standard) oder ASYNCHRON bearbeitet werden soll.

Bei Synchroner-Bearbeitung wird die verwendete Verbindung erst nach Abschluss der Bearbeitung der Service-Funktion mit dem entsprechenden Ergebnis an den Aufrufer zurückgegeben.

Bei langlaufenden Service-Funktionen kann es von Vorteil sein, diese Asynchron ausführen zu lassen. Dabei wird direkt nach der Eingangsprüfung ein Asynchron-Handle zurückgegeben. Mit diesem Asynchron-Handle kann dann zu einem späteren Zeitpunkt das Ergebnis für die Service-Funktion abgerufen werden.

Ist das Ergebnis für die Service-Funktion nicht wichtig, bzw. soll zu einem späteren Zeitpunkt kein Ergebnis abgerufen werden, so kann mit der Vorgabe

WWSVC-EXECUTE-MODE=ASYNCHRON\_NO\_RESULT

die Service-Funktion ohne Asynchron-Handle abgesetzt werden. Damit ist dann kein weiterer Zugriff für diese Service-Funktion mehr notwendig.

Mögliche Parameter für WWSVC-EXECUTE-MODE

- SYNCHRON (\*DEFAULT wird immer gesetzt wenn nicht angegeben/unerlaubter Wert)
- ASYNCHRON
- ASYNCHRON\_NO\_RESULT

Es ist zu beachten das der Asynchron-Modus nur erlaubt ist, wenn dies für die Service-Funktion bzw. das WWSVC-System entsprechend aktiviert wurde. Ist die Asynchrone Ausführung nicht erlaubt, so wird der Zugriff automatisch SYNCHRON ausgeführt.

Es ist möglich eine maximale Vorhaltezeit nach Abarbeitung der Service-Funktion für das Ergebnis vorzugeben. Dabei wird nach Erreichen der maximalen Vorhaltezeit der Asynchron-Handle ungültig und ein Abrufen des Ergebnisses ist dann für die Service-Funktion nicht mehr möglich.

Bei Asynchroner Bearbeitung wird ein Zugriffs-Schlüssel für den späteren Zugriff zurückgegeben. Dabei ist zu beachten das hier jeweils ein Cookie WWSVC-ASYNCHRON-HANDLE, sowie ein Attribut in der COM-RESULT-JSON Struktur WWSVC\_ASYNCHRON\_HANDLE zurückgegeben wird.

Bei ASYNCHRON erhält man dabei den Zugriffs-Handle, bei ASYNCHRON\_NO\_RESULT wird der Standard-Wert ASYNCHRON-ACCEPTED zurückgegeben, da kein weiterer Zugriff mehr notwendig ist.

## WWSVC-CURSOR Beispiel Abruf

Im folgenden wird die Verwendung eines WWSVC-Cursor beschrieben.

### Funktionsaufruf mit WWSVC-Cursor-Anforderung

Aufruf einer normalen WWSVC-Funktion mit Angabe von WWSVC-CURSOR=CREATE sowie der für den ersten Abruf maximal gewünschten Anzahl Sätze:

```
PUT /WWSVC/EXECJSON/ HTTP/1.1
Host: meineWebware.de
WWSVC-EXECUTE-MODE: SYNCHRON
WWSVC-HASH: fca2429bfa267aca2a4d3668f0f3f4c2
WWSVC-CURSOR: CREATE
WWSVC-TS: Mon, 17 Oct 2016 06:58:43 GMT

{"WWSVC_PASSINFO":{
  "SERVICEPASS":"c1d9136edf6424c828ae2d282ec262cb",
  "GET_RESULT_MAX_LINES":2,
  "GET_WWSVC_CURSOR":"CREATE",
  }...
```

### WWSVC-Cursor Ergebnis nach erstem Erfolgreichem Abruf

Sind für die Abfrage mehr als die gewünschte GET\_RESULT\_MAX\_LINES Sätze vorhanden, so wird ein WWSVC-Cursor erstellt und im HTTP-Header-Feld WWSVC-Cursor zurückgegeben.

```
HTTP/1.1 200 OK
Content-Length: 39239
Content-Type: application/json
WWSVC-CURSOR=WWSVC-06B20260-CURSOR

{"COMRESULT": {"STATUS....
```

### Folgeabruf eines WWSVC-Cursors

Unter Angabe des WWSVC-CURSOR werden die nächsten GET\_RESULT\_MAX\_LINES Sätze gelesen.

```
PUT /WWSVC/EXECJSON/ HTTP/1.1
Host: meineWebware.de
WWSVC-EXECUTE-MODE: SYNCHRON
WWSVC-HASH: fca2429bfa267aca2a4d3668f0f3f4c2
WWSVC-CURSOR: WWSVC-06B20260-CURSOR

{"WWSVC_PASSINFO":{
  "SERVICEPASS":"c1d9136edf6424c828ae2d282ec262cb",
  "GET_RESULT_MAX_LINES":2,
  "GET_WWSVC_CURSOR":" WWSVC-06B20260-CURSOR ",
  }...
```

**WWSVC-Cursor Beenden bei Lesen letztem Satz**

Wird der letzte Satz in einem WWSVC-Cursor gelesen, so wird der WWSVC-Cursor automatisch Beendet, und dies der Kennung WWSVC-CURSOR: **CLOSED** an den Client zurückgemeldet.

```
HTTP/1.1 200 OK
Content-Length: 39239
Content-Type: application/json
WWSVC-CURSOR: CLOSED
```

```
{"COMRESULT": {"STATUS
```

```
}...
```

**WWSVC-Cursor Schließen vom Client aus**

Wird ein CURSOR nicht mehr benötigt so muss dieser aktiv vom Client geschlossen werden. Nach Schließen eines Cursor werden die Ressourcen in der WWSVC32.exe Anwendung freigegeben. Hierzu gibt es eine extra Funktion die über einen eigenen Zugangspunkt angesprochen wird.

```
GET /WWSVC/WWSERVICE/CURSORCLOSE/123SVP789/WWSVC-12345678-CURSOR HTTP/1.1
Host: myDomain.de
WWSVC-TS:
WWSVC-REQID:
WWSVC-APPID:
```

Im Beispiel oben ist 123SVP789 der gültige ServicePass welcher für die Erstellung des Cursor verwendet wurde. WWSVC-12345678-CURSOR ist der gültige WWSVC-CURSOR

Ist das Beenden erfolgreich so erhält man den HTTP-Code 200

```
HTTP/1.1 200 OK
Content-Length: 85
Content-Type: text/html
Date: Tue, 17 Mar 2015 23:53:42 GMT
Server: WebWare PoWeR Server

{"COMRESULT":
  {
    "STATUS": 200,
    "CODE": "200 OK",
    "INFO": "WWSVC-CURSOR CLOSED"
  }
}
```

**Fehler beim Abruf von WWSVC-CURSOR**

Tritt beim Abruf eines Problem auf so wird dies mit dem HTTP-Status-Code 404 sowie einer weiteren Fehlernummer kenntlich gemacht. Ein Fehler tritt dann auf wenn ein WWSVC-CURSOR im WW-Server nicht bekannt ist, bzw. in der Zwischenzeit ungültig geworden ist.

## WWSVC-DYNOBJ Dynamisches Objekt Beispiel Abruf

Im folgenden wird die Verwendung eines Dynamischen Objekt Token beschrieben. Ein Dynamischer Objekt-Token wird automatisch vom WWSVC32.exe Prozess erstellt falls ein Dynamisches Objekt wie zum Beispiel eine MEMTAB erzeugt wird.

### Antwort auf die Erstellung eines Dynamischen Objektes

Zuvor wurde zum Beispiel MEMTAB.EXEC mit Parameter INIT aufgerufen. Es wird ein Dynamischer Objekt Token erzeugt welcher im Antwort HTTP Header Feld WWSVC-DYNOBJ zurückgegeben wird. Ein Dynamischer Objekt Token ist immer in der Form WWSVC-12345678-DYNOBJ angegeben.

```
HTTP/1.1 200 OK
WWSVC-DYNOBJ: WWSVC-094669B0-DYNOBJ

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

### Zugriff mit gültigem Dynamischen Objekt Token

Wurde ein Dynamischer Objekt Token erstellt, so muss beim Zugriff auf die erstellen Dynamischen Objekte dieser mit angegeben werden, um das Routing auf den zugehörigen Prozessor-Knoten sicher zu stellen. Der WWSVC-DYNOBJ Token kann dabei als HTTP-Header Feld oder mit Höherer Priorität als JSON Element des ServicePasses mit übergeben werden.

```
PUT /WWSVC/EXECJSON/ HTTP/1.1
WWSVC-EXECUTE-MODE: SYNCHRON
WWSVC-REQID: 353
WWSVC-DYNOBJ: WWSVC-0947ABB8-DYNOBJ

{"WWSVC_PASSINFO":
{"SERVICEPASS": "3b2ee14877f3cdf77ea9e2c571ccd3e7",
"GET_RESULT_MAX_LINES": 2,
"GET_RESULT_TYPE": "JSON",
"WWSVC_DYNOBJ": "WWSVC-0947ABB8-DYNOBJ"
...}
```

### Zugriff mit ungültigem Dynamischen Objekt Token

Wird mit einem ungültigen, also entweder unbekannten oder durch den Prozessablauf oder Inaktivitätsüberschreitung ungültig gewordenen Dynamischen Objekt Token zugegriffen so erhält man einen HTTP 404 Fehlercode als Antwort. Sind die

```
HTTP/1.1 404 Resource not found
WWSVC-DYNOBJ: WWSVC-06BF74C8-DYNOBJ@CLOSED

{"COMRESULT": {"STATUS": 404, "CODE": "404 Resource not found",
"INFO": "Resource not found",
"ERRORCODE": 50610,
"ERRORLINK": "DOCWWSVC/ERR.HTML/#50610",
"ERRORINFO": "WWSVC-DYNOBJ WWSVC-06BF74C8-DYNOBJ is not known"}}
```

Haben Sie die Detailltiefe für die Fehlercodes im System-Cockpit aktiviert so erhalten Sie hier auch einen erweiterten Fehlercode aus dem Bereich 50610 bis 50614 der den genauen Zustand/Fehler erklärt.

Wird ein Dynamische Objekt Token (WWSVC-DYNOBJ) ungültig so wird dieser im HTTP-Header mit der Erweiterung @CLOSED markiert.



## Übersicht der Fehler/Status-Codes

### HTTP-Status-Codes

Folgende Status-Codes werden vom WWSVC im WW-Server verwendet. Dabei werden die vordefinierten Status-Codes von HTTP 1.1 verwendet.

#### Status - OK

Code	Info	Details
200	OK	Aktion wurde erfolgreich ausgeführt
202	ACCEPTED	Aktion wurde erfolgreich ausgeführt, es sind noch weitere Aktionen notwendig. Beispiel: Service-Pass wurde registriert, muss jedoch noch von Administrator freigegeben werden.  Beispiel: Es wurde eine Asynchrone Anfrage abgesetzt (WWSVC_EXECUTE_MODE=ASYNCHRON) , bzw. eine Asynchrone Anfrage ist noch nicht fertig bearbeitet.

#### Status-Fehler

Code	Info	Details
400	BAD REQUEST	Anfrage ist Fehlerhaft, wird abgelehnt
401	AUTHORIZATION REQUIRED	Es ist eine Anmeldung für den Zugriff erforderlich, der Client muss Benutzer und Passwort für die Secured-App angeben.  In der SecuredApp kann eine Benutzer-Liste festgelegt werden, welche bei Registrierung und/oder Zugriff geprüft wird.  Ist eine Benutzer-Liste für den EXEC-Zugriff eingetragen, so wird bei jedem EXEC-Aufruf ohne gültigen SESSION-TOKEN der Code 401 zurückgegeben. Für einen gültigen SESSION-TOKEN ist ein CONNECT-Aufruf erforderlich.
403	FORBIDDEN	Zugriff auf diese Funktion ist nicht erlaubt
404	NOT FOUND	Die Angeforderte Funktion ist nicht gültig. Beispiel: es wird mit einem Service-Pass auf die WEWARE-Services zugegriffen, welcher nicht bekannt ist.
405		
406	NOT ACCEPTABLE	Die Anfrage kann nicht angenommen werden. Beispiel: Versuch der Registrierung eines Servicepasses, aber WEBWARE-Services sind deaktiviert
409	CONFLICT	Der Zugriff ist wegen eines Conflictes mit einem anderen Zugriff nicht möglich

**Server-Status-Fehler**

Code	Info	Details
500	INTERNAL SERVER ERROR	Bei der Bearbeitung der Service-Anfrage ist ein interner Fehler im WEBWARE-Server aufgetreten, die Anfrage konnte nicht bearbeitet werden.
501	NOT IMPLEMENTED	Die angeforderte Funktion ist vom WEBWARE-Service nicht implementiert

## Liste der Rückgabe-Codes

Wird eine Service-Funktion ausgeführt, so kann im Fehlerfall eine Zusatzinformation ausgegeben werden, welche nähere Hinweise und Details zu dem Bearbeitungszustand enthält. Wenn Sie diese Codes erhalten wollen, so muss der Detailgrad für die COMRESULT-Struktur auf mindestens 1 gesetzt werden. Dieser kann im System-Cockpit unter aktiviert und gesetzt werden.

### INFO 0 .. 9999

Code	Bereich	Details
0	ALL	Ausführung OK
200	GETASYNCREULT	Ausführung OK, Ergebnis wird übergeben. Asynchron-Handle ist nun ungültig.
200	VALIDATE	Servicepass wurde erfolgreich geprüft und Zugriff ist erlaubt
202	GETASYNCREULT	Service-Funktion wird asynchron abgearbeitet. Ergebnis ist noch nicht verfügbar.

### WARNUNGEN 10000 .. 49999

Code	Bereich	Details
10000	REGISTER	Service-Pass wurde registriert, wartet noch auf Freigabe von Administrator

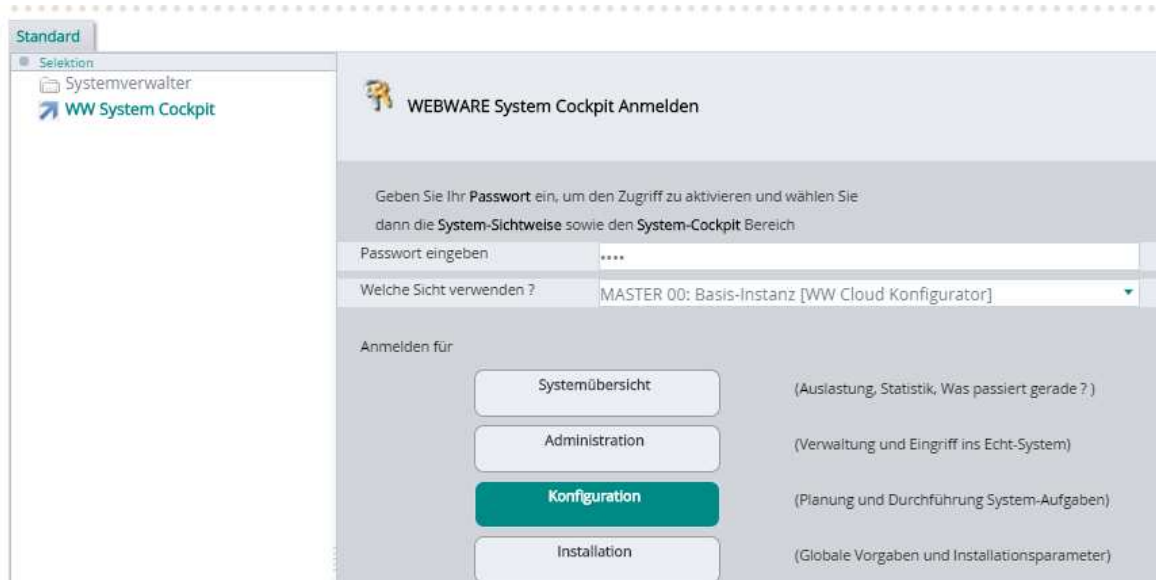
### FEHLER 50000 .. 99999

Code	Bereich	Details
50000	SERVER	Anfrage war nicht zu verarbeiten, Verarbeitung abgebrochen
50100	REGISTER	Servicepass nicht zu registrieren, da Anwendung nicht bekannt
50101	REGISTER	Servicepass nicht zu registrieren, da Anwendung für Registrierung gesperrt
50102	REGISTER	Servicepass nicht zu registrieren, da Zugriff aus diesem Netzsegment verboten ist
50103	REGISTER	Servicepass nicht zu registrieren, da für die SecuredApp eine Benutzer-Liste für die Registrierung aktiviert ist, und kein Benutzer/Passwort, bzw. kein passendes übergeben wurde.
50200	VALIDATE	Servicepass nicht gefunden
50300	CONNECT	Fehler beim CONNECT mit einem Service-Pass, Benutzer/Passwort nicht in Benutzer-Liste vorhanden, oder ServicePass ist nicht lauffähig.
50301	CONNECT	Fehler ServicePass ist nicht erlaubt
50302	CONNECT	Fehler ServicePass ist nicht freigeschaltet.
50400	EXEC GETASYNCREULT	Session-Token für Zugriff ist nicht gültig. Rückgabe Code 401 Authentifizierung erforderlich (CONNECT..) um einen gültigen WWSVC-SESSION-TOKEN zu erstellen.
50500	GETASYNCREULT	Zugriff ist nicht erlaubt. Asynchron-Handle ist nicht gültig

50600	WWSVC-CURSOR NOT KNOWN	Der übergebene WWSVC-Cursor ist nicht gültig und kann nicht ausgeführt werden
50601	WWSVC-CURSOR NOT VALID	Der übergebene WWSVC-Cursor ist nicht mehr gültig und kann nicht mehr ausgeführt werden. In der Zwischenzeit wurde der verarbeitende Knoten der den WWSVC-CURSOR ausgeführt hatte ungültig.
50602	WWSVC-CURSOR TIME-OUT	Der übergebene WWSVC-Cursor ist in der Zwischenzeit durch überschreiten der maximalen Haltezeit für den WWSVC-Cursor ungültig geworden.
50603	WWSVC-CURSOR NOT ALLOWED	Die Verwendung von WWSVC-Cursor ist Aufgrund von System-Vorgaben nicht gestattet
50610	WWSVC-DYNOBJ NOT KNOWN	Der übergebene WWSVC-DYNOBJ Identifier ist nicht gültig und kann nicht ausgeführt werden
50611	WWSVC-DYNOBJ NOT VALID	Der übergebene WWSVC-DYNOBJ ist nicht mehr gültig und kann nicht mehr ausgeführt werden. In der Zwischenzeit wurde der verarbeitende Knoten der den WWSVC-DYNOBJ ausgeführt hatte ungültig.
50612	WWSVC-DYNOBJ TIME-OUT	Der übergebene WWSVC-DYNOBJ ist in der Zwischenzeit durch überschreiten der maximalen Haltezeit für den WWSVC-DYNOBJ Zugriffstoken ungültig geworden.
50613	WWSVC-DYNOBJ INVALID CURSOR	Es wurde bei der Übergabe eines WWSVC-DYNOBJ ein WWSVC-CURSOR Objekt angegeben, welches nicht zum WWSVC-DYNOBJ Kompatibel ist. (Unterschiedlicher Pool/Prozess-Knoten)
50614	WWSVC-DYNOBJ NOT ALLOWED	Die Verwendung/Erstellung von Dynamischen Objekten ist Aufgrund von System-Vorgaben nicht gestattet

## Aktivieren der WEBWARE Services im SystemCockpit

Um das WEBWARE Service System WWSVC für Ihr WEBWARE System zu aktivieren, melden Sie sich mit der Option Konfiguration an.



## Aktivieren von WWSVC

Sie finden die wichtigsten Parameter zur Aktivierung der WEBWARE Services (WW-SVC) im Bereich Sicherheits-Center. Sie können hier mit dem ersten Parameter die WW-SVC aktivieren. Mit den 4 weiteren Parametern können Sie die Zugriffs- und Freigabevorgaben für Inter- und Intranet vorgeben.



Weitere Parameter finden Sie unter "System Konfiguration" in dem Eintrag WWSVC WEBWARE Services Einträge.

Standard		
Selektion	Daten	
	Beschreibung	Systemwert
Systemverwalter		
WWSC Konfiguration 0-Basis-Instanz	WW Service System Aktiv	1
System Konfiguration	WW Service über Internet erlaubt	1
System Information	WW Service über Intranet erlaubt	1
System Basis Konfiguration	WW Service Internet immer Admin Freigabe	1
Programmpfade	WW Service Intranet immer Admin Freigabe	1
Netzwerk Anbindung	Servicefunktion Wiederholzeit in Sekunden bei Restart WWSVC	30
Logon/Anmelde Vorgaben	Setze diesen Mandant wenn keine Vorgabe,0=Standard	1
RAR-Server Konfiguration	Starte WWSVC nur für diesen Mandanten, 0=Alle	1
Programm/Module Definition	Detail-Level COMRESULT-Stuktur, 0=Aus bis 9=Voll	3
System Sperrzeiten	WW Service CORS für Originn erlauben	N
Zeitgesteuerte Aufgaben	WW Service CORS für alle Domains erlauben	J
WW-TAPI Vorgaben	WW Service asynchrone Zugriffe erlauben	1
WWCC Konfiguration	WW Service asynchrone Ergebnisse Vorhaltezeit (Sekunden)	20
WW eMail Messaging System	Service-Zugang erlaubt Standard Netzwerk Zugang	1
WWSVC WEBWARE Services	Service-Zugang erlaubt Zusatz Netzwerk Zugang	1
WW-Programm Versionen	Funktionsnamen aus HTTP-Befehlen ergänzen	1
	Setze RESULT_MAX_LINES bei 0 auf	100
	Servicefunktion Wie oft versuchen bei Restart WWSVC	2
	WW Service WEB Proxy erlauben	1
	CURSOR Verwendung erlaubt	1
	CURSOR Beenden nach Sekunden Inaktivität	20
	CURSOR Zu einem Zeitpunkt nur 1x pro ServicePass	1
	DYNOBJ Dynamische Objekte Verwendung erlaubt	1
	DYNOBJ Beenden nach Sekunden Inaktivität	40
	DYNOBJ Zu einem Zeitpunkt nur 1x pro ServicePass	1

## Die Bedeutung der Konfiguration Einträge

### WW Service System Aktiv

Ist dieser aktiviert, so wird beim nächsten Start des WEBWARE-Servers das WW-SVC Subsystem mit hochgefahren, und ist verwendbar.

### WW Service über Internet erlaubt

Ist dieser Schalter aktiviert, so ist der Zugriff mit Service-Pässen aus dem Internet heraus erlaubt. Vorgabe welcher Bereich Intranet und welcher Bereich Intranet ist wird in der Konfiguration im Bereich "Konfiguration" > "System Konfiguration" > "Logon/Anmelde Vorgaben" > "Inter-Net bzw. Intra-Net" gemacht.

### WW Service Internet immer Admin Freigabe

Meldet sich eine Service-Anwendung an Ihrem WEBWARE System an so haben Sie die Möglichkeit Zugriffe für die Service-Pässe auf Hersteller/Anwendungsebene ohne Freigabe durch Administrator freizugeben. Ist dieser Schalter hier gesetzt, so muss beim Erstzugriff von Service-Pässen aus dem Internet immer ein Administrator diesen nach der Registrierung freigeben.

### WW Service über Intranet erlaubt

Mit diesem Schalter können sie den Zugriff auf WEBWARE Services aus dem Intranet heraus erlauben.

**WW Service Intranet immer Admin Freigabe**

Meldet sich eine Service-Anwendung an Ihrem WEBWARE System an so haben Sie die Möglichkeit Zugriffe für die Service-Pässe auf Hersteller/Anwendungsebene ohne Freigabe durch Administrator freizugeben. Ist dieser Schalter hier gesetzt, so muss beim Erstzugriff von Service-Pässen aus dem Intranet immer ein Administrator diesen nach der Registrierung freigeben.

**Servicefunktion Wiederholzeit in Sekunden bei Restart**

Wird eine Servicefunktion in einem WWSVCxx-Knoten für die Ausführung eingeplant, und kommt es zu einem Restart/Abbruch des WWSVCxx-Knoten, so kann hier die Zeit in Sekunden bis zum letzten Versuch die Servicefunktion erneut einzuplanen vergehen soll.

**Setze diesen Mandant wenn keine Vorgabe, 0=Standard-Mandant**

Falls keine Mandanten-Vorgabe beim Zugriff von Service-Pässen vorhanden ist, so kann hier ein Mandant angegeben werden, der für den Zugriff auf die WWSVCxx-Knoten verwendet wird.

**Starte WWSVC nur für diesen Mandanten, 0=Alle**

Bis die Konfiguration des WWSVC-Systems vollständig möglich ist, kann man mit diesem Schalter den Start der WWSVCxx-Knoten auf einen einzelnen Mandanten beschränken, bzw. mit 0 die WWSVCxx-Knoten für jeden Mandanten aktivieren. Diese Option ist ab Version 09.02.2016 ohne Funktion.

**Detail-Level COMRESULT-Struktur 0=Aus bis 9=Voll**

Hier kann angegeben werden wie viel Informationen / Hinweise in der COMRESULT-Rückgabe Struktur vom WEBWARE-Server WWSVC übergeben werden sollen.

- Level 0: (DEFAULT)  
Rückgabe von STATUS, CODE, INFO
- Level 1:  
Rückgabe von STATUS, CODE, INFO, ERRORCODE
- Level 2:  
Rückgabe von STATUS, CODE, INFO, ERRORCODE, ERRORLINK
- Level 3:  
Rückgabe von STATUS, CODE, INFO, ERRORCODE, ERRORLINK, ERRORINFO

weitere Level's werden später noch definiert.

Level 8: Zeit-Debug-Modus

Es wird eine Struktur mit Informationen über das Laufzeitverhalten mit zurück gegeben

Level 9: Voll-Debug-Modus

Es wird eine Struktur für die interne Funktionsabarbeitung im Prozessor-Knoten mitgegeben.

***Was bedeuten die einzelnen COMRESULT-Werte:***

STATUS: Status-Code (HTTP-Status-Code siehe Tabelle weiter oben)

CODE: Standard-Beschreibung HTTP-Status-Code

INFO: Zusätzlicher Beschreibender Text vom WEBWARE WWSVC System

ERRORCODE: Interner Fehler/Hinweis/Warnungs-Code der auf die Dokumentation verweist

siehe weiter oben in Liste der Rückgabe-Codes..

ERRORLINK: Link auf die HTML-Dokumentation in Ihrem WW-Server zu dem ERRORCODE

ERRORINFO: Zusätzlicher Beschreibender Text zu dem ERRORCODE

### ***Kurze Einführung in CORS***

Wird eine Verbindung von einem Browser zu Ihrem WEBWARE-Server System aufgebaut, und die Basisanwendung im Browser wurde nicht von Ihrem WEBWARE-System aufgerufen, so kommt es im Browser zu einem Fehler, da eine Verbindung mit Ajax und anderen Web-Technologien nur zu dem Ursprungs-Server von die Basisanwendung aufgerufen wurde erlaubt ist. Diese Browser-Interne Sicherheits-Restriktion kann mit Hilfe von CORS umgangen werden. CORS steht für Cross-Origin Resource Sharing, also frei übersetzt.. "Zugriff auf Ressourcen auch von anderen Ursprungs-Rechner erlauben".

Bei CORS muss der aufgerufene WEB-Server (in diesem Falle Ihr WEBWARE-Server) die Erlaubnis und auch die Reichweite für diese Verbindung angeben. Dabei wird angegeben wie mit den Sitzungsparametern, Cookies (HTTP-Header) und den Daten umgegangen werden soll. Dürfen diese nur bei Verbindungen zu Ihrem WEBWARE-Server, oder auch bei anderen Verbindungen verwendet werden.

### **WW Service CORS für Origin erlauben**

Ist dieser System-Wert aktiviert, so ist CORS nur für diese Verbindung zu Ihrem WEBWARE-Server gültig. Sitzungsparameter dürfen vom Browser nicht an andere Domains weitergegeben werden.

### **WW Service CORS für alle Domains erlauben**

Ist dieser System-Wert aktiviert, so ist CORS für alle Verbindungen auch zu Ihrem WEBWARE Server gültig und die Parameter für die Verbindung zu Ihrem WEBWARE-Server dürfen auch mit anderen Systemen geteilt werden. (nicht so gut..)

### **WW Service ASYNCHRONE Zugriffe erlauben**

Mit diesem System-Wert kann der Asynchrone Zugriff auf Ihre WEBWARE-Services aktiviert werden. Bei Asynchronen Zugriffen wird das Ergebnis des Zugriffs zwischengespeichert und kann später vom Initiator der Service-Funktion abgerufen werden.

### **WW Service ASYNCHRONE Ergebnisse Vorhaltezeit (Sekunden)**

Mit diesem System-Wert kann angegeben werden wie lange Ergebnisse einer Asynchron ausgeführten Service-Funktion maximal im System zum ersten Abruf bereit gehalten werden. Die Angabe erfolgt in Sekunden. Bei Vorgabe von 0 gibt es keine zeitliche Begrenzung für das Vorhalten.

### **Service-Zugang erlaubt Standard-Netzwerk Zugang**

Mit dieser Option können Sie festlegen ob der Zugriff auf die WWSVC WEBWARE WEB Services über die Standardschnittstelle Ihrer WEBWARE-Instanz erlaubt ist. Bei Deaktivierung ist der Zugriff auf WWSVC-Funktionen über `https://[Server-Name]/WWSVC/..` nicht erlaubt und es wird eine 404er Fehlermeldung zurückgegeben.

### **Service-Zugang erlaubt Zusatz-Netzwerk Zugang**

Mit dieser Option können Sie festlegen ob der Zugriff auf die WWSVC WEBWARE WEB Services über die Zusatzschnittstelle Ihrer WEBWARE-Instanz erlaubt ist. Bei Deaktivierung ist der Zugriff auf WWSVC-Funktionen über `https://[Server-Name]/WWSVC/..` nicht erlaubt und es wird eine 404er



Fehlermeldung zurückgegeben. Diese Einstellung macht nur Sinn wenn Sie auch eine Zusatz-Schnittstelle im Bereich Konfiguration > System-Konfiguration > Netzwerk Anbindung > WEB Schnittstelle eingetragen haben.

### Funktionsnamen aus HTTP-Befehlen ergänzen

Ist dieser Parameter aktiviert, so wird beim Zugriff auf Datenobjekte (Bspl: Artikel) ohne Angabe von Funktionen des Datenobjekts (Bspl: ARTIKEL.UPDATE, ARTIKEL.GET) der Funktionsname anhand des HTTP-Befehls ermittelt und zur Laufzeit für das Datenobjekt vorgegeben.

Details:

Um Funktionen auszuführen haben Sie die Möglichkeit diese per normalen HTTP PUT/POST auszulösen. Dabei ist zu Beachten das der WWSVC-Server eingehende Funktionsnamen bei Zugriff auf eine Datenobjekt wie zum Beispiel ARTIKEL, BELEGE, (...) oder WORKFLOW je nach HTTP-Befehl auf festgelegte Funktionen routet. Dies geschieht jedoch nur wenn dieser Parameter hier aktiv ist.

Wird bei einem Funktionsaufruf direkt eine Funktion für ein Datenobjekt angegeben so entfällt das weitere Routen. (ARTIKEL.GET wurde angegeben so wird keine Funktion über HTTP gesetzt)

Beispiel (zur Vereinfachung im EXECURL Format):

Wenn Sie eine Funktion ausführen auf das Datenobjekt ARTIKEL und geben dabei keine weitere Ausführungs-Funktion an wie zum Beispiel (ARTIKEL.UPDATE, ARTIKEL.INSERT, ARTIKEL.DELETE ARTIKEL.GET), dann wird anhand des HTTP-Befehls die Ziel-Funktion automatisch angefügt.

HTTP	URL	Ausführungsfunktion
PUT	/WWSVC/SVP/ARTIKEL	ARTIKEL.UPDATE
POST	/WWSVC/SVP/ARTIKEL	ARTIKEL.INSERT
GET	/WWSVC/SVP/ARTIKEL	ARTIKEL.GET
INSERT	/WWSVC/SVP/ARTIKEL	ARTIKEL.INSERT
DELETE	/WWSVC/SVP/ARTIKEL	ARTIKEL.DELETE

Beachten Sie das bei Ausführung mit dem GET-Befehl teilweise keine Übergabe von JSON-Parametern aus Browsern im Body-Bereich erfolgt.

Sie können also für den Zugriff auf die /WWSVC/EXEC... Funktionen folgende HTTP-Befehle verwenden:

- PUT entspricht Datenobjekt . UPDATE
- POST entspricht Datenobjekt . INSERT
- GET entspricht Datenobjekt . GET
- INSERT entspricht Datenobjekt . INSERT
- DELETE entspricht Datenobjekt . DELETE
- UPDATE entspricht Datenobjekt . UPDATE
- EXEC entspricht Datenobjekt . EXEC (Ausführen von Workflows usw.)
- OPTIONS Abfragen der API-JSON Struktur für das angegebene Objekt

### Setze RESULT\_MAX\_LINES bei 0 auf (System-Wert)

Wird bei einer Service-Funktion der Parameter RESULT\_MAX\_LINES nicht übergeben bzw. ist dieser 0 so wird diese System-Wert gesetzt um durch Fehlerhafte Abfragen keine Laufzeitprobleme auszulösen. Gerade bei großen Datenbeständen wird bei den Rückgabewerten sehr schnell mehrere 100 MB erreicht so dass die Übertragung zwischen den Systemen (WWS und Client) überdurchschnittlich lange dauert.

Werden mehr Sätze benötigt so kann dies der WWSVC-Schnittstelle durch HTTP-Header (WWSVC-ACCEPT-RESULT-MAX-LINES) bzw. JSON Attribut (GET\_RESULT\_MAX\_LINES ) mit angegeben werden.

## **Verbindungsrouting von Browser zu Fremdsystem /WWSVC/WWSERVICE/WEBPROXY**

../WWSVC/WWSERVICE/WEBPROXY/Ziel-URL

Ist ein Zugriff aus dem Browser durch CORS nicht möglich, so bietet der WEBWARE-Server die Möglichkeit der Einrichtung eines HTTPS-Proxy's mit dem einzelne Verbindungen zu Fremdsystemen aufgeführt werden können.

Wenn zum Beispiel Ihre Anwendung vom WEBWARE Server ausgeliefert wird und Sie versuchen aus dem Browser auf andere Domain's zu zugreifen, so wird das im Browser unter Umständen unterbunden. Dies liegt dann daran das die "andere" Domain also der andere WEB-Server keine Erlaubnis zum Verbindungsaufbau gibt wenn Ihre Anwendung nicht von diesem anderen WEB-Server geladen wurde.

Nähere Hinweise zu CORS finden Sie "Kurze Einführung in CORS".

Um den WEBWARE WWSVC-WEB-Proxy zu aktivieren muss neben WWSVC auch der System-Wert

WW Service WEB Proxy erlauben

1

im Bereich Konfiguration>System Konfiguration>WWSVC WEBWARE Services ../aktiviert werden.

Der WWSVC WEBPROXY hat die Funktion die Daten von Ihrer Browser Anwendung zu Empfangen. Dabei wird beim Aufruf, also in der URL die gewünschte ZIEL-URL in einem speziellen Format übergeben. Das Format ist notwendig damit das HTTP-Protokoll nicht gestört wird.

Neben der ZIEL-URL wird der Inhalt (Body) komplett sowie einige Header-Felder aus der Anfrage übernommen.

### **Beispiel eines Zugriffs auf Afterbuy:**

Hierbei wird die Interface-Adresse

<https://api.afterbuy.de/afterbuy/ABInterface.aspx>

in

<https%5E%60%60api.afterbuy.de%60afterbuy%60ABInterface.aspx>

umgesetzt. Dabei werden die Doppelpunkte ":" mit der Kennung %5E sowie die Verzeichnistrenner "/" mit der Kennung %60 ersetzt.

## **Mögliche Rückgabe-Codes**

400 Wenn die Übergabeparameter Fehlerhaft sind, (Kein HTTPS, usw.) bzw. keine Ausführung

404 Wenn WEBPROXY nicht erlaubt ist, bitte System-Wert überprüfen.

Sonstige Fehler- / Statuscodes kommen dann vom entfernten System

**Beispiel Afterbuy Abruf****Abruf einer Afterbuy-Schnittstellen-XML:**

```
POST /WWSVC/WWSERVICE/WEBPROXY/https%5E%60%60api.afterbuy.de%60afterbuy%60ABInterface.aspx HTTP/1.1
Host: meine-webware.de
Connection: keep-alive
Content-Length: 395
Pragma: no-cache
Cache-Control: no-cache
Origin: https://meine-webware.de
Content-Type: application/xml
Accept: */*
DNT: 1
Referer: https://meine-webware.de /wwapp-test/index.html
Accept-Encoding: gzip, deflate, br
Accept-Language: de-DE,de;q=0.8,en-US;q=0.6,en;q=0.4

<?xml v.....
```

**Aus dieser Anfrage wird dann folgender HTTPS Aufruf erzeugt und entsprechend abgesetzt:**

```
POST /afterbuy/ABInterface.aspx HTTP/1.1
Content-Length: 395
Content-Type: application/xml
Host: api.afterbuy.de
Referer: https://api.afterbuy.de/afterbuy/ABInterface.aspx
Pragma: no-cache
Cache-Control: no-cache

<?xml v.....
```

**Empfangen der Antwort und Rückgabe an Ihre Anwendung**

Die empfangenen Daten des entfernten Rechners werden dann ohne weitere Bearbeitung als Antwort an Ihre Anwendung zurückgegeben..

```
HTTP/1.1 200 OK
Cache-Control: private
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
Strict-Transport-Security: max-age=31536000; includeSubDomains
Set-Cookie: ASP.NET_SessionId=vvjj0fkdjgt3otukzkzpxasw; path=/; HttpOnly
X-Powered-By: ASP.NET
X-Frame-Options: SAMEORIGIN
Date: Tue, 07 Jun 2016 07:16:30 GMT

6cb4
<?xml version="1.0" encoding="utf-8"?>
<Afterbuy>
  <CallStatus>Success</CallStatus>
```

**CURSOR Verwendung erlaubt**

Mit diesem Schalter kann die Verwendung von CURSOR beim Zugriff auf Datenobjekte erlaubt werden. Im Standard ist die CURSOR Verwendung erlaubt. Bei einem Cursor ist es möglich "Seiten-Weise" auf ein Ergebnis zuzugreifen.

Bei Verwendung eines CURSOR wird der Zugriff direkt auf den erzeugenden Prozessknoten für diesen CURSOR beschränkt und Folgezugriffe auf das Bestehende Datei-/Datenobjekt weitergereicht so dass man ohne Angabe von einem Startdatensatz direkt mit dem nächsten Datensatz arbeiten kann.

Ein Cursor ist so lange gültig bis er entweder vom Client gelöscht wurde, der Zeitpunkt der Inaktivität erreicht wurde, oder aber der betreffende Prozess-Knoten abgeschaltet wurde.

**CURSOR Beenden nach Sekunden Inaktivität**

Da ein CURSOR einen höheren Speicherverbrauch auf einem Prozess-Knoten auslöst, da hier Datenobjekte speziell für diesen CURSOR vorgehalten werden, kann man über diesen Sekunden Zeitwert steuern ab welcher Inaktivitätszeit der CURSOR automatisch gelöscht wird. Bei dem Wert 0 wird keine Zeitprüfung angewendet. Bei Werten größer 0 wird nach jedem Zugriff der Zeitpunkt berechnet ab dem der CURSOR nicht mehr gültig ist. Wird dieser Zeitpunkt erreicht so wird der CURSOR gelöscht. Der Client erhält dann eine 404 Fehlermeldung beim Zugriff auf den Cursor die wie in der Fehlertabelle beschrieben eine Extrafehler-Code von 50600 bis 50603 abhängig vom CURSOR Zustand.

**CURSOR: Zu einem Zeitpunkt nur 1x pro ServicePass**

Es ist möglich für einen ServicePass zur Laufzeit mehrere CURSOR gleichzeitig zu öffnen, und offen zu Halten. Dies kann gewünscht sein, kann aber auch als Angriffsvektor verwendet werden um gezielt einzelne Prozess-Knoten, die normalerweise für den Client nicht erkennbar sind, zu Überlasten.

Über diesen Systemwert kann man die Anzahl der Cursor pro Service-Pass und Zeitpunkt auf einen CURSOR begrenzen. Ist dieser Systemwert aktiv, so wird bei Erzeugen eines neuen CURSOR der bisher aktuelle automatisch geschlossen und als Ungültig markiert.

**DYNOBJ Dynamische Objekte Verwendung erlaubt**

Bei Verwendung von Objekten die zur Laufzeit erzeugt werden, wie zum Beispiel MEMTAB, ist es notwendig das diese länger als der eigentliche Servicefunktionen Zugriff leben. Mit Hilfe von Dynamischen Objekt-Token kann dies erreicht werden. Dieser wird bei Bedarf automatisch vom WWSVCxx.EXE Prozess erstellt und bei HTTP-Header Feld an den Client zurückgegeben. Will man nun auf ein solches Dynamisches Objekt zugreifen, muss der Dynamische Objekt Token als HTTP-Header oder JSON-Wert mitgegeben werden. Werden in der Folge mehrere Dynamische Objekte erstellt (mehrere MEMTAB/Selektionspool, usw.) so werden diese unter dem gleichen bereits bestehenden Dynamischen Objekt Token zusammengefasst und angesprochen. Wird das letzte Dynamische Objekt eines Dynamischen Objekt Token entfernt, so wird dieser automatisch ungültig und entfernt. Der Client erhält dann den Dynamischen Objekt-Token mit dem Zusatz "@CLOSED" übergeben.

Bspl: WWSVC-DYNOBJ=WWSVC-ABCDEFGH-DYNOBJ@CLOSED

Mit diesem Systemwert kann die Verwendung von Dynamischen Objekten erlaubt/verboten werden. Standard ist Erlaubt.

**DYNOBJ Beenden nach Sekunden Inaktivität**

Da ein Dynamisches Objekt Token einen höheren Speicherverbrauch auf einem Prozess-Knoten auslöst, da hier Dynamische Datenobjekte speziell für diesen Objekt vorgehalten werden, kann man über diesen Sekunden Zeitwert steuern ab welcher Inaktivitätszeit der Dynamische Objekt Token automatisch gelöscht wird. Bei dem Wert 0 wird keine Zeitprüfung angewendet. Bei Werten größer 0 wird nach jedem Zugriff der Zeitpunkt berechnet ab dem der Dynamische Objekt Token nicht mehr gültig ist. Wird dieser Zeitpunkt erreicht so wird der Dynamische Objekt Token gelöscht. Der Client erhält dann eine 404 Fehlermeldung beim Zugriff auf den Dynamische Objekt Token die wie in der Fehlertabelle beschrieben eine Extrafehler-Code von 50610 bis 50614 abhängig vom Dynamischen Objekt Token Zustand.

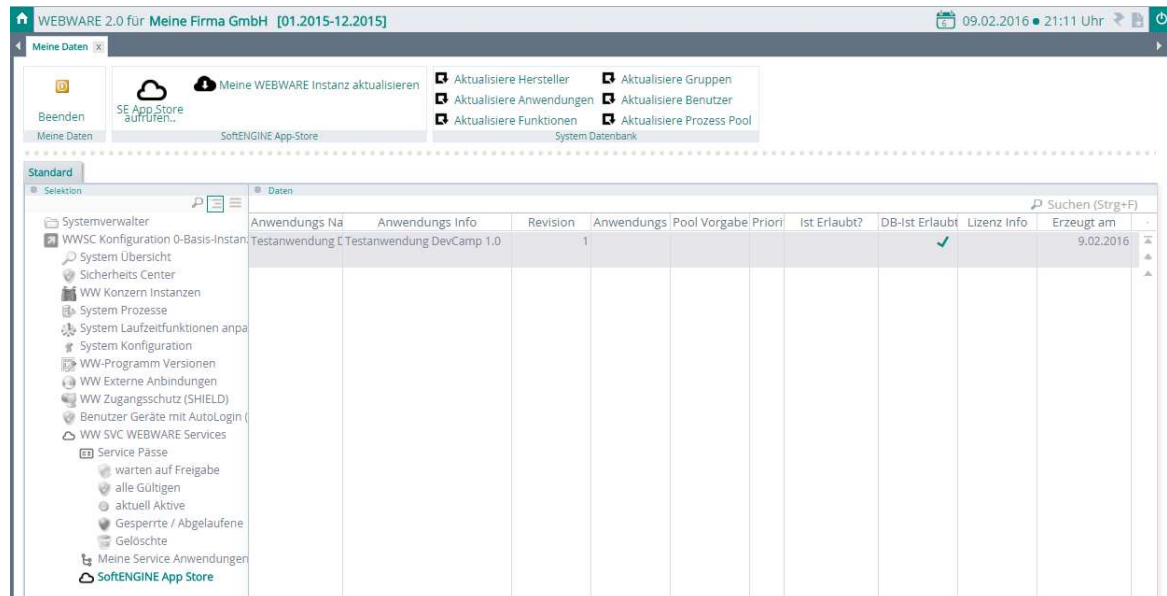
**DYNOBJ Zu einem Zeitpunkt nur 1x pro ServicePass**

Es ist möglich für einen ServicePass zur Laufzeit mehrere Dynamische Objekt Token gleichzeitig zu öffnen, und offen zu Halten. Dies kann gewünscht sein, kann aber auch als Angriffsvektor verwendet werden um gezielt einzelne Prozess-Knoten, die normalerweise für den Client nicht erkennbar sind, zu Überlasten.

Über diesen Systemwert kann man die Anzahl der Dynamischen Objekt Token pro Service-Pass und Zeitpunkt auf einen begrenzen. Ist dieser Systemwert aktiv, so wird bei Erzeugen eines neuen Dynamischen Objekt Token der bisher aktuelle automatisch geschlossen und als Ungültig markiert.

## Verwaltung von WWSVC-Elementen

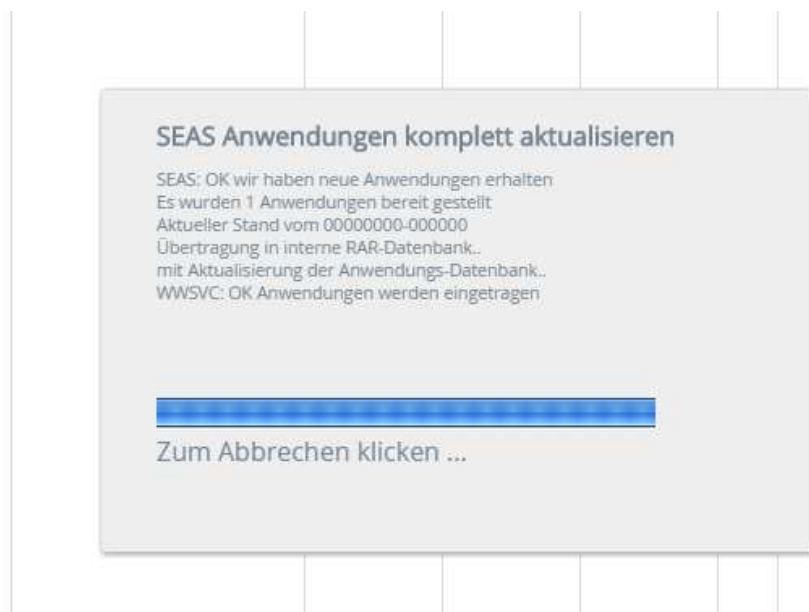
Hier finden Sie Informationen welche Elemente im WW-SVC System vorkommen, und wie diese verwaltet werden. Sie finden im System-Cockpit in den Bereichen Administration/Konfiguration im Baum einen Eintrag "WW SVC WEBWARE Services".



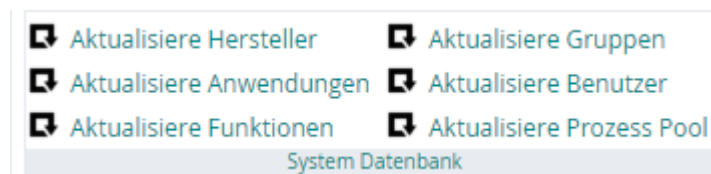
Unterhalb des Eintrages "WW SVC WEBWARE Services" finden Sie die 3 Haupt-Objekte die für die WEBWARE-Services verwendet werden.

- SoftENGINE AppStore
- Meine Service-Anwendungen
- Service-Pässe

Um Ihrem WEBWARE-Server System neue Hersteller/Anwendungen bekannt zu machen, können Sie diese aus dem SoftENGINE AppStore laden ("Meine WEBWARE Instanz aktualisieren"). Dabei werden neue Hersteller sowie deren Anwendungsbeschreibungen im Datenbankbereich (RAR-Server/DB) aktualisiert.



Nach der Aktualisierung werden dann die Information im RAR-Bereich in der Datenbank abgelegt. Mit Hilfe der "System-Datenbank Funktionen" können Sie dann ein sofortiges Aktualisieren der WW-Server System-Cockpit Datenbank auslösen.



Die Liste der erlaubten Anwendungen wird dabei im SoftENGINE AppStore verwaltet und kann von dort aus verändert werden.



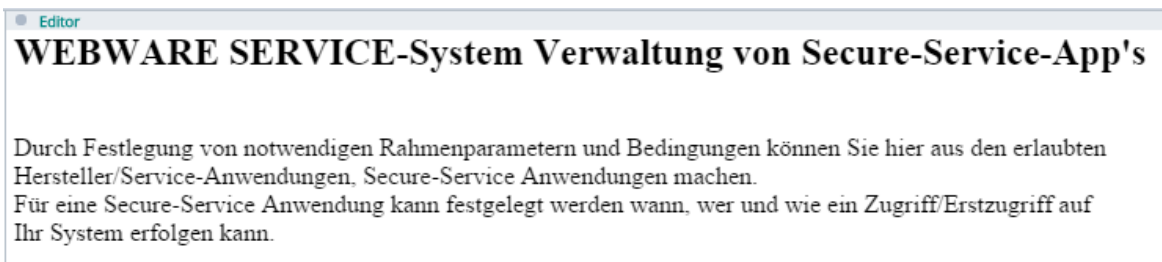
## Service Anwendungen (SecuredApp)



Wurde eine Hersteller-Anwendung freigegeben, so erscheint diese im Baum im Bereich "Service-Anwendung verwalten".

Eine Service Anwendung enthält Rahmenparameter und Zugriffsbeschränkungen mit denen Sie diese an Ihre Sicherheitsvorgaben anpassen können.

Unterhalb dieses Astes finden Sie für jeden freigegebenen Hersteller einen Eintrag, sowie unter jedem Hersteller dessen freigegebene Hersteller-Anwendungen.



Wird ein Hersteller unterhalb von "Service-Anwendungen verwalten" ausgewählt, so erhalten Sie eine Liste der Secured-App's die Sie aus dem Bereich "Hersteller-Wendungen verwalten" für Ihr WEBWARE-Server WWSVC-System freigegeben haben.



Aus dem Menü können Sie die Änderung von Vorgabeparameter für Ihre Secured-App vornehmen, oder auch Benutzer-Listen für die Zugriffs-Steuerung auf Ihr System unter Verwendung dieser Secured-App verwalten.

### Folgende Parameter einer Service-Anwendung sind zu Bearbeiten



## WWSVC Service Anwendung anlegen

Hersteller	SoftENGINE GmbH
Anwendung	Testanwendung DevCamp 1.0
Zugriffs-ID	2
Version	1
Mandant für Ausführung	2 : Beispieldaten
Begrenze Zeitraum	0 : Aktueller Zeitraum
Register Modus	1 : Erlaubt, Freigabe durch Admin
Benutzerliste bei Registrierung	<input checked="" type="checkbox"/>
Gruppe für Registrierung	1 : GRUPPE-1
Ausführungs Modus	1 : Ausführung erlaubt
Benutzerliste bei Ausführung	<input checked="" type="checkbox"/>
Gruppe für Ausführung	2 : GRUPPE-2
Max.Sitzungsdauer(Sec.)	20000
IntraNet nicht starten	<input type="checkbox"/>
InterNet nicht starten	<input type="checkbox"/>
Nur in NetSecureArea	<input type="checkbox"/>
Standardeintrag	<input checked="" type="checkbox"/>
Erlaubt von StartUhrzeit	<input type="text"/>
Erlaubt bis StartUhrzeit	<input type="text"/>
Setze Ablaufdatum	<input type="text"/>
Setze Ablauf in Tagen	<input type="text"/>

#### **Zugriffs-ID**

Die Zugriffs-ID ist ein eindeutiger Wert, welcher beim Speichern einer Service-Anwendung automatisch ermittelt wird. Bei der Registrierung eines Client's für diese Service-Anwendung muss die Zugriffs-ID angegeben werden, um eine Auswahl für Mandant und Belegzeitraum sowie die hier angegebenen Konfigurationswerte zu Erreichen.

#### **Mandant für Ausführung**

Geben Sie hier den Mandanten vor der bei der Ausführung der Service-Funktion verwendet wird.

#### **Begrenze Zeitraum**

Wählen Sie hier den Belegzeitraum welcher für die Ausführung der Service-Funktion verwendet wird. Im Standard wird hier der Belegzeitraum 0 (Aktuelles Jahr) verwendet. Falls Sie Daten in abweichende Belegzeiträume **schreiben** wollen, müssen Sie hier den entsprechenden Zeitraum wählen.



**Registrierungs-Modus**

Register Modus	0 : Gesperrt, keine Registrierung
Benutzerliste bei Registrierung	0 : Gesperrt, keine Registrierung
Gruppe für Registrierung	1 : Erlaubt, Freigabe durch Admin
Ausführungs Modus	2 : Erlaubt, Freigabe automatisch
Benutzerliste bei Ausführung	9 : Gelöscht: nicht mehr Registrierbar
Gruppe für Ausführung	

Sie können hier vorgeben ob diese Anwendung Registrierung Gesperrt oder Erlaubt ist. Bei erlaubten Anwendungen können Sie festlegen ob ein Administrator die Anwendung manuell freigeben muss.

**Benutzerliste bei Registrierung [verwenden]**

Sie können hier angeben ob bei Registrierung für einen Service-Pass die Benutzerprüfung anhand einer Benutzer-Liste erfolgen muss. Ist dieser Parameter aktiviert, so wird bei der Registrierung der Benutzer-Name sowie das übergebene Benutzer-Passwort gegen die Benutzer-Liste die bei der Secured-App hinterlegt ist geprüft. Wird der Benutzer mit dem übergebenen Passwort nicht in der Benutzer-Liste gefunden, so wird der Service-Pass nicht erstellt und die Registrierung abgelehnt.

**Benutzer-Gruppe für Registrierung [wer darf registrieren]**

Hier können Sie aus der Benutzer-Gruppen Verwaltung der WWSVC eine Benutzergruppe für die Registrierung vorgeben. Bei Vorgabe und Aktivierung von "Benutzerliste bei Registrierung" können sich nur noch Benutzer anmelden die mit Benutzer-Kennung und Benutzer-Passwort in der Benutzer-Gruppe eingetragen sind.

**Ausführungs Modus**

Ausführungs Modus	1 : Ausführung erlaubt
Benutzerliste bei Ausführung	0 : Gesperrt, keine Ausführung erlaubt
Gruppe für Ausführung	1 : Ausführung erlaubt
Max.Sitzungsdauer(Sec.)	9 : Gelöscht: nicht mehr ausführbar
IntraNet nicht starten	

Sie können hier vorgeben ob die Ausführung für registrierte Service Anwendungen gesperrt, oder erlaubt ist. Ist die Ausführung gesperrt, so können auch gültige Service-Pässe nicht auf diese Anwendung zugreifen.

**Benutzerliste bei Ausführung [verwenden]**

Sie können hier angeben ob bei Verwendung eines Service-Passes bzw. die Benutzung einer Secured-App zuvor eine Benutzer-Anmeldung (Authentifizierung) mit Benutzer und Passwort gegen die bei der Secured-App hinterlegten Benutzer-Liste erfolgen muss. Ist dieser System-Wert aktiviert so erhält der Aufrufer die Fehlermeldung "401-Authorization required" (Benutzer-Anmeldung erforderlich) wenn er ohne erfolgreiche Anmeldung seinen Service-Pass verwendet.

Die Anmeldung muss, bei Aktivierung dieses System-Wertes, über die WWSVC/WWSERVICE/CONNECT-Funktion erfolgen. Dabei wird Service-Pass + Benutzer-Name + Benutzer-Passwort übergeben, und der Aufrufer erhält bei Erfolg einen [zeitlich beschränkten] Sitzungs-Schlüssel übergeben, der bei jedem weiteren Zugriff übergeben werden muss.

***Benutzer-Gruppe für Ausführung[wer darf ausführen]***

Hier können Sie aus der Benutzer-Gruppen Verwaltung der WWSVC eine Benutzergruppe für die Ausführung vorgeben. Bei Vorgabe und Aktivierung von "Benutzerliste bei Ausführung" können nur noch Benutzer Funktionen für diese Service-Anwendung ausführen die mit Benutzer-Kennung und Benutzer-Passwort in der Benutzer-Gruppe eingetragen sind.

***Benutzerliste maximale Sitzungsdauer [in Sekunden]***

Wird eine Benutzerliste für den Zugriff verwendet, so erhält der Client bei erfolgreicher Anmeldung (Authentifizierung) einen Sitzungs-Schlüssel. Hier können Sie nun angeben wie lange dieser Sitzungsschlüssel maximal gültig sein soll.

0 : Keine zeitliche Begrenzung, nach einmaliger Anmeldung bleibt der Sitzungsschlüssel immer gültig, welcher bei jedem Zugriff zusätzlich angegeben werden muss.

>0: Zeit in Sekunden bis zum Ablauf des Sitzungsschlüssels. Der Sitzungsschlüssel muss immer mit angegeben werden. Ist der Sitzungsschlüssel nicht mehr gültig, so erhält der Client die Fehlermeldung "401- Authorization required" Anmeldung erforderlich als Fehler-Code

***IntraNet / InterNet nicht starten***

Sie können mit diesen 2 Parametern festlegen ob der Zugriff aus Inter-/Intranet verboten ist.

***Nur in NetSecureArea***

Hier können Sie einen IP-Adress-Bereich vorgeben für den sicheren Betrieb. Wenn hier ein Wert angegeben wird, so können nur Client's aus dem erlaubten Bereich diese Anwendung starten.

***Standardeintrag***

Hier können Sie die Version der Service-Anwendung auf die aktuelle Beschränken, bzw. den Zugriff auch von älteren Versionen erlauben (Wird später noch genauer implementiert)

***Erlaubt von / bis Uhrzeit***

Hier können Sie den Zugriff auf einen Zeitbereich am Tag einschränken.

***Begrenze auf Mandant/Zeitraum***

Hier können Sie den Mandant der für diese Service-Anwendung verwendet wird vorgeben, sowie den Belegzeitraum.

***Setze Ablaufdatum***

Hier können Sie eine Datum setzen, bis zu diesem der Zugriff von gültigen Service-Pässen für diese Anwendung erlaubt ist.

***Setze Ablauf in Tagen***

Hier können Sie ein Ablaufterm in Tagen setzen ab wann der Zugriff von gültigen Service-Pässen für diese Anwendung erlaubt ist.

## Verwalten von Benutzer-Listen



Mit dem Menü-Befehl "Benutzerlisten verwalten" können Sie die Benutzerverwaltung für WWSVC aufrufen. Die Datenhaltung erfolgt dabei im RAR-Datenbankbereich. Sie haben hier die Möglichkeit Benutzergruppen anzulegen, sowie Benutzer mit Passwort dort je Gruppe zu verwalten.

Wenn Sie in Ihrer "Service Anwendung", Vorgaben für die Verwendung von Benutzer-Listen gemacht haben, so ist der Zugriff, also Registrierung und/oder Verwendung der "Service Anwendung" für einen Service-Pass nur möglich wenn der Client den korrekten Benutzer-Name und Benutzer-Passwort übergibt.



Je nachdem ob Sie eine Benutzergruppe für Registrierung und/oder Ausführung bei einer Service-Anwendung hinterlegt haben, wird im Menü der entsprechende Befehl mit angezeigt. Damit können Sie direkt die Benutzer-Listen verwalten.

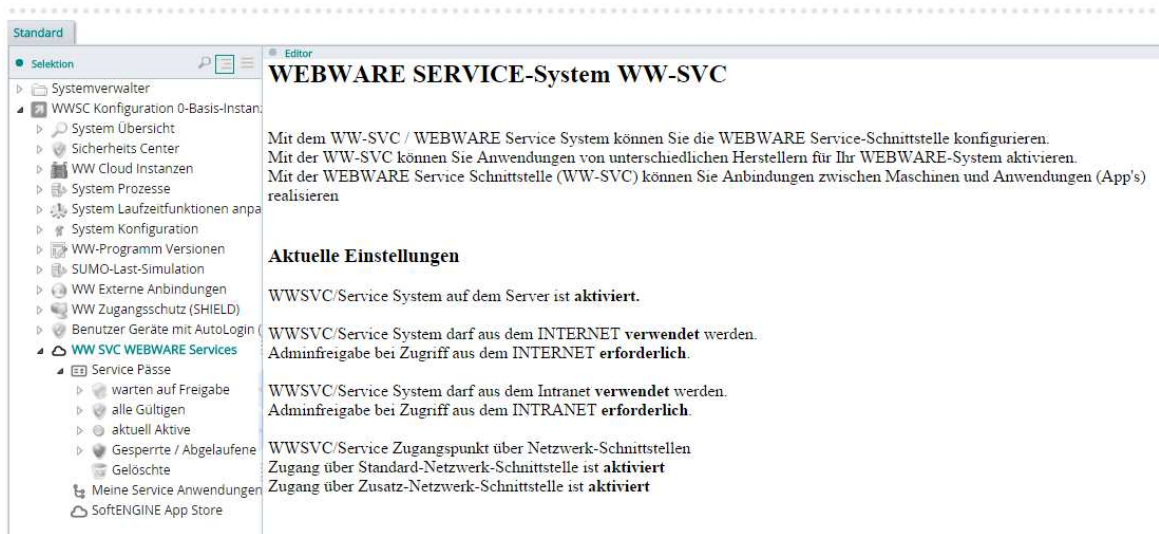
## Verwalten von Benutzer-Listen im Bereich Administration

Wenn Sie sich im Bereich Administration anmelden, können Sie dort im Bereich Zugangs-Verwaltung unter WW-SVC-Services BenutzerListen für die freigegebenen Service-Anwendungen die zugehörigen Benutzerlisten verwalten.



## Service Pässe

Will ein Client auf Ihr WEBWARE System zugreifen, so benötigt er einen gültigen Service-Pass. Hierzu muss er wie weiter oben beschrieben sich bei Ihrem WEBWARE System registrieren. Je nach Vorgaben



ist eine automatische Registrierung sowie eine manuelle Freigabe vom System-Administrator möglich. Ist der Zugriff erlaubt, so erhält der Client einen Service-Pass mit dem er von nun an auf Ihr WEBWARE System und der damit verbundenen Secured Service Anwendung zugreifen kann.

Die Service Pässe werden nach Ihrem Status im Baum dargestellt.

## Warten auf Freigabe

Neue Service-Pässe die nicht automatisch freigegeben werden sollen erscheinen in diesem Bereich. Der Administrator hat die Möglichkeit diese zu Löschen (Alle, Löschen älter wie heute, Einzel-Löschung), sowie den ServicePass freizugeben, also für den Zugriff zu aktivieren.



Wir ein Service-Pass angeklickt, so können Sie folgende Parameter verwalten:



Hersteller	SoftENGINE C++ WEBSER
Anwendung	SE-Anwendung-1
Ablaufdatum	
Ablaufuhrzeit	
Diesen Pool verwenden:	
Dieser Mandant	2
Dieser Belegzeitraum	0 : Aktueller Zeitraum

### ***Abaufdatum***

Datum bis zu welchem der Service-Pass gültig ist

### ***Ablaufuhrzeit***

Ist das Ablaufdatum gesetzt, so wird der Eintrag ab der entsprechenden Uhrzeit gesperrt.

### ***Diesen Pool verwenden***

Hier können Sie einen Prozess-Pool angeben, in dem eingehende Service-Funktionen für diesen Service-Pass verarbeitet werden sollen.

### ***Dieser Mandant***

Hier wird der Mandant ihres WEBWARE Sytemes vorgeben auf welchen dieser Service-Pass zugreifen darf/kann.

### ***Dieser Belegzeitraum***

Hier wird der Belegzeitraum vorgegeben auf welchen der Service-Pass zugreifen kann.

### Alle Gültigen Service-Pässe / Alle aktiven Service-Pässe

Hier finden Sie eine Übersicht über alle gültigen ServicePässe, bzw. unterhalb je Hersteller diese sortiert. Sie können hier einzelne ServicePässe Sperren, bzw. komplett Löschen. Es ist auch möglich alle Service-Pässe der Liste zu Löschen bzw. nur Service-Pässe bis auf den heutigen Tag.

WEBWARE 2.0 für Meine Firma GmbH [01.2015-12.2015]

Meine Daten x

Beenden  
Meine Daten

Alle Löschen  
Löschen nicht von Heute

Löschen/Entfernen  
Sperren/Quarantäne

Standard

Selektion

- Systemverwalter
- WWSC Konfiguration 0-Basis-Instanz
- WW SVC WEBWARE Services
  - Service Pässe
    - warten auf Freigabe
    - alle Gültigen
    - SoftENGINE C++ WEB
    - aktuell Aktive
    - Gesperrte / Abgelaufene
    - Gelöschte
  - Service Anwendungen verwa
  - Hersteller Anwendungen ver

Suchen (Strg+F)

Anwendungs	ServicePass ID
SE-Anwendung-1	d8d3effa7a19b2a50f5a949985e4f84a

### Gesperrte/Abgelaufene Service-Pässe

Hier können Sie gesperrte bzw. abgelaufene ServicePässe verwalten und bei Bedarf diese wieder Freigeben bzw. komplett Löschen. Es ist auch möglich alle Service-Pässe der Liste zu Löschen bzw. nur Service-Pässe bis auf den heutigen Tag.

WEBWARE 2.0 für Meine Firma GmbH [01.2015-12.2015]

Meine Daten x

Beenden  
Meine Daten

Alle Löschen  
Löschen nicht von Heute

Löschen/Entfernen  
Freigeben

Standard

Selektion

- Systemverwalter
- WWSC Konfiguration 0-Basis-Instanz
- WW SVC WEBWARE Services
  - Service Pässe
    - warten auf Freigabe
    - alle Gültigen
    - aktuell Aktive
    - Gesperrte / Abgelaufene
    - Gelöschte
  - Service Anwendungen verwa
  - Hersteller Anwendungen ver

Suchen (Strg+F)

Anwendungs	ServicePass ID
SE-Anwendung-1	d8d3effa7a19b2a50f5a949985e4f84a



## Referenz der Servicefunktionen

Eine Referenz aller Service-Funktionen und die zugehörigen Datenressourcen finden Sie im Handbuch

WW-WEBSERVICE-FUNKTIONEN.PDF

## Übersicht Änderungen an diesem Dokument

Änderungsdatum	Änderungsgrund/Erweiterung
Rel 1.0: 06.03.2015	Erweiterung der URL-Schnittstelle. Möglichkeit der Übergabe von Parameter-name. Beispiel: PARA1="Ich bin ein Text", damit wird Parmeter PARA1 definiert mit Inhalt "Ich bin ein Text". Erweiterung COMRESULT um ERRORCODE/ERRORLINK/ERRORINFO Schalter COMRESULT-Detail im System-Cockpit eingebaut EXECURL baut nun einen JSON-String Richtung WWSVC_Prozessor auf Kommunikation zwischen WWS und WWSVC32 wurde geändert.
Rel 1.1 10.06.2015	Erweiterung um CONNECT Funktion für Benutzerlisten für Service-Pässe
Rel 1.2 22.06.2015	Überarbeitung des Dokumentes, Neustrukturierung und Ergänzung/Erweiterung sowie Aufnahme neuer Funktionen für Benutzer-Listen und Beschreibung der Neuerungen im System-Cockpit
Rel 1.3 12.08.2015	Asynchron Modus integriert. Damit kann man nun auch Abfragen Asynchron in WWSVC übergeben, und das Ergebnis zu einem späteren Zeitpunkt abrufen.
Rel 1.4 21.08.2015	Secure-APP-ID eingeführt um WWSVC-Anwendungen lokal im WW-System konfigurieren zu können, sowie SEAS
Rel 1.5 01.10.2015	Beschreibung der Registrierung bei SEAS
Rel 1.6 18.01.2016	Bei Session-Connect werden nun bei Fehlerhaftem ServicePass, oder ServicePass ohne Freigabe der Fehlercode 406 zurückgegeben.
Rel 1.7 11.02.2016	Überarbeitung und Aktualisierung der Dokumentation für das erste DEV-Camp WW SVC
Rel 1.8 11.02.2016	Referenz Servicefunktionen aktualisiert.
Rel 1.9 11.02.2016	Einbau von Funktionen anfordern in iWWSVC.JS
Rel 2.0 19.02.2016	Erweiterung der HTTP-Schnittstelle (PUT/POST/GET/INSERT/DELETE/OPTIONS) und Strukturierung der Zugriffsfunktionen für Klassen A.UPDATE, A.INSERT, A.GET, A.DELETE, A.OPTIONS Neue Systemwerte um Zugriff auf WWSVC auf einzelne Netz-Zugangspunkte zu Begrenzen. Neuer Systemwert um Funktionsnamen ausgehend vom HTTP-Befehl zu setzen
Rel 2.1 24.02.2016	Neuer Optionaler HTTP-Header für Service-Pass hinzugefügt
Rel 2.2 09.03.2016	Anpassung an neue iWWSVC.js, Auslagerung von Service-Funktionen in eigenes Handbuch WW-WEBSERVICE-FUNKTIONEN.PDF
Rel 2.3 19.04.2016	Neuer System-Wert im System-Cockpit um bei keiner Angabe von maximale Anzahl Ergebniszeilen diese automatisch zu Begrenzen.
Rel: 2.4 07.06.2016	WWSVC Proxy (WEBPROXY) Schnittstelle implementiert und dokumentiert. WWS REV: 12628
Re.: 2.5 09.09.2016	Löschfunktion für Service-Pässe auch für Aktive/Gesperrte usw.
Rel: 2.6 13.09.2016	Erweiterung der Laufzeitinformationen für Pool, Service-Funktionen sowie Prozesse im Bereich Administration und Konfiguration  REV: WWS 12652
Re.: 2.7 17.10.2016	Erweiterung der WWSVC Curosr Implementierung



Rel: 2.8 21.12.2016	Erweiterung CURSOR (Time-Out Steuerung und Systemwerte) DYNObj Dynamische Objekte, Aktualisierung der Dokumentation  Aktualisierung WWSVC-js auf 1.10 vom 21.12.2016
---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------