

↗ SoftENGINE

↗ UPDATE

TECHNIK



Webservices

Ab Version WEBWARE 2.01 (Erprobungsphase 2016)



Philipp Müller, kaufmännischer Leiter der
Hund-Katze-Goldfisch GmbH

Zukünftig wollen wir in verschiedenen Bereichen im Unternehmen auf die WEBWARE setzen und müssen natürlich unseren Webshop auch entsprechend anbinden.

In der BüroWARE haben wir es bislang über Standardschnittstellen & Datenbank-Zugriff direkt gelöst, in der WEBWARE helfen uns zukünftig Webservices. Diese werden von unserer IT selbst programmiert. Es geht also darum: Wie komme ich an Daten, die in der WEBWARE gespeichert sind und wie bekomme ich Daten hinein in die WEBWARE?

Wie kann beispielsweise unser Kunde online den aktuellsten Lagerbestand des gewünschten Artikels einsehen oder wie können Stammkunden im Kundenkonto Aktionspreise und Sonderpreise ersehen, die sich natürlich von den öffentlichen ShopPreisen unterscheiden?

Da die Preispflege komplett in der

WEBWARE hinterlegt ist, kann unser Bestandskunde über einen Webservice-Call genau seinen Preis für diesen Artikel, an diesem Tag mit der gewünschten Menge einsehen. Die ganze Preisermittlung läuft mit Formeln und über Individualisierungen in der WEBWARE ab. Im Shop sieht der Kunde jedoch tatsächlich SEINEN aktuellen Preis. Was ebenfalls unschlagbar effizient ist, ist die Tatsache, dass die Online-Bestellungen direkt in unsere WEBWARE gelangen.

Und das alles ohne Synchronisation nach bestimmten Zeitintervallen. Da die SoftENGINE Webservices direkt integriert sind, haben sie erhebliche Geschwindigkeitsvorteile gegenüber Third Party Lösungen.

Auch unseren Außendienst haben wir über WEBWARE Webservices angebunden. Mit einer externen APP auf den Tablets unserer Mitarbeiter kann mittels der Webservices DIREKT und absolut SICHER auf die zur Verfügung gestellten Daten unserer WEBWARE zugegriffen werden.

Das bedeutet in erster Hinsicht, dass wir InHouse Mitarbeiter im normalen

Tagesgeschäft unsere Stammdaten bearbeiten und pflegen, spezielle Adresskonditionen anpassen u.v.m. und ohne dass eine Datenaufbereitung stattfinden muss, alle gespeicherten Daten in der WEBWARE, per Webservices sofort auch dem Außendienst zur Verfügung stehen.

Technische Details

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 https://myDomain.de/WWSVC
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 <code>../WWSVC/[PASS-ID]/Funktions-Name..</code>
Parameter	Parameter können direkt beim Aufruf in der URL (URL-Encoded) übergeben werden. <code>../WWSVC/[PASS-ID]/MeineFunktion/10001/SORT=KNDNR</code> 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	Entwickler-Interface Test-Implementierung für WWSVC mit Javascript . Test: https://meine-webware.de/svc-js/svcjs.html

Zugriffsarten für WEBSERVICES:

SYNCHRON

Ergebnis der Service-Funktion wird mit der gleichen Verbindung zurückgegeben

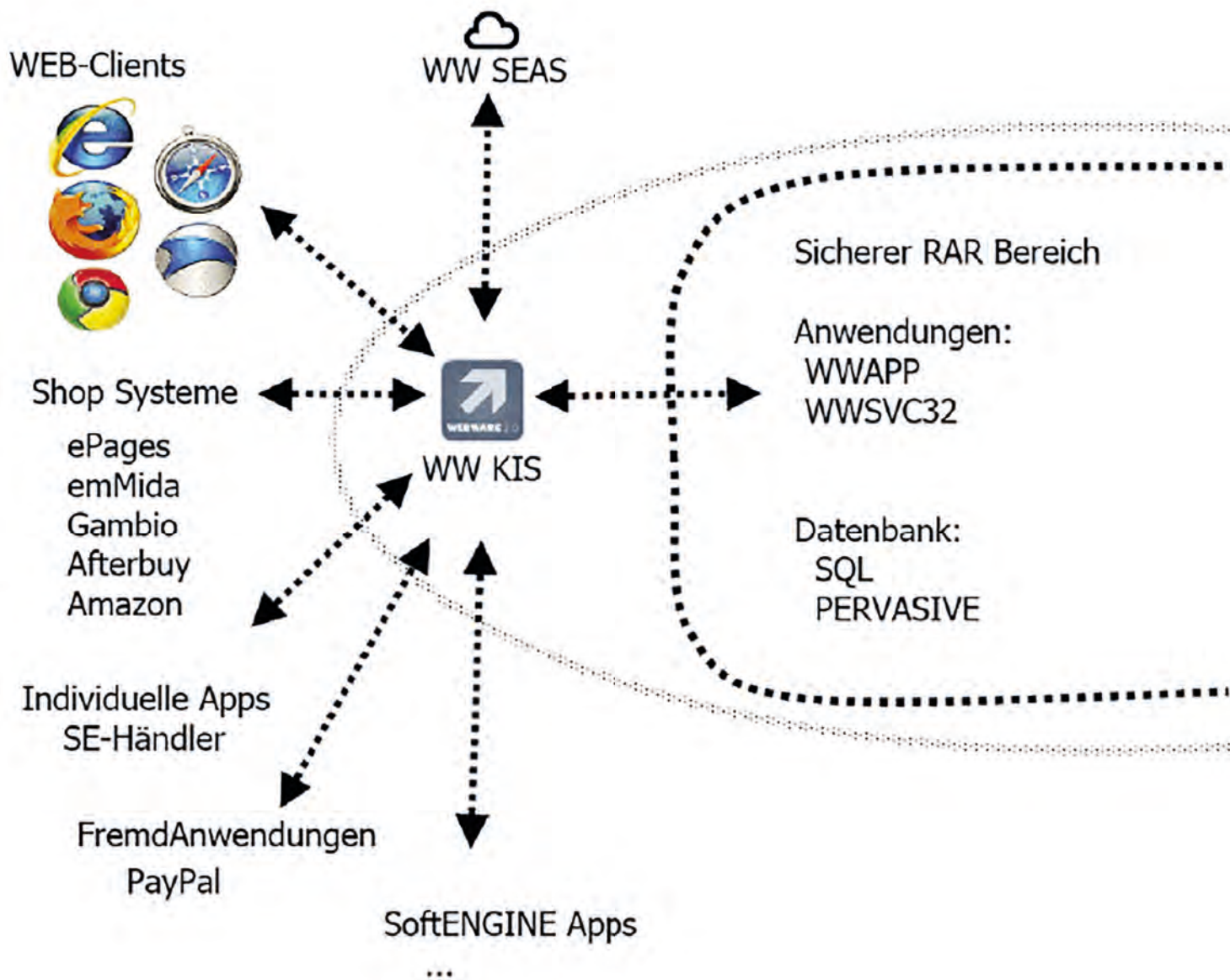
ASYNCHRON

Service-Funktion wird im 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 höchsten Wert gelegt.

Die Verwaltung von Hersteller/Anwendungen (WWSVC-Apps) wird im SoftENGINE AppStore SEAS vom SoftENGINE Partner 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.

WEBWARE iWWSVC.js

Beschreibung des Interfaces für WWSVC Javascript in der Version 1.03

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\SVC-JS\

Zielbeschreibung

Das hier vorliegende Javascript Interface soll Ihnen helfen die WWSVC Schnittstelle der WEBWARE WEB Services (WWSVC) zu verstehen und im Browser anzuwenden.

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

[svcjs-html/svcjs.css/testapp.js](#)

Neben der HTML/CSS Datei für die Oberflächen Gestaltung finden Sie in der Datei „TESTAPP.JS“ die Anwendung der jWWSVC.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 drei Rahmen dargestellt, Anwendungen, Service-Pass und Funktionen. Durch Klick auf das PLUS-Symbol können Sie jeweils neue Elemente entsprechend des Rahmens anlegen.

WWSVC Test Anwendung 1.0 / 22.04.2016

Active	Hersteller	Anwendung					Options	
<input checked="" type="radio"/>	ca06a64216d69j5132a7650e7646603l	ba9e79087e55df02d1dke8762883f7b7	Edit	Delete	Register	Abrufen		

Service-Pass	Session-Token	Pass	Pass	Session	Options
2e8e75d5e85d6ecc1dfd95e0f91e7c3b					Validate DeRegister Anfordern Abrufen

Active	Function			Exec	ExecExtra	Exec	ExecExtra	Options			
<input checked="" type="radio"/>	ADRESSARTIKEL	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	ADRESSE	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	ANSPRECHPARTNER	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	ARTIKEL	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	BELEG	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	BELPOS	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	CHARGE	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	GET_RELATION	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	LIEFERADRESSE	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	PROJEKT	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	PUT_RELATION	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	SERIENNUMMER	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE
<input checked="" type="radio"/>	TERMIN	Edit	Delete	Sync	Sync	Async	Async	Abrufen	INSERT	UPDATE	DELETE

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 14 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 drei 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 Eingang 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 drei 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 zwei 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, dass 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 zwei 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 zwei 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, AsyncMaxTimeToPoll)

Die Ausführung erfolgt dann mit der Funktion

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

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 Ressource.

WWSVC_GET_FUNCTION_OPTIONS (AppHndl, „BELEGE“, aCB, opt_FuncVer)

E) 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, dass je nach Funktionsaufruf nicht alle Felder gefüllt werden.

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

CBP.RESULTCODE	0 Fehler v1 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 Anwendungszugriff....

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.

DEREGISTER

Wird ein Service-Pass nicht mehr benötigt, so kann mit der DEREGISTER-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.

GETASYNCRESULT

Mit dem GetAsyncResult-Zugriff kann das Ergebnis für eine asynchron aufgebene 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.

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.

- **HERSTELLER** 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
- **PASSWORT** 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/53f69160a5b0b89136ba1c6390c1e5d1/04abf1c38b8522869f857dcffa3c5500/1//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.

Das vollständige TECHNIK UPDATE „Webservices“ (172 Seiten) finden Sie online: <http://www.softengine.wiki/3516/>

SoftENGINE

Kaufmännische Softwarelösungen
für Handel, Industrie & E-Commerce

Alte Bundesstraße 16 • 76846 Hauenstein
Telefon: +49 (0) 63 92 - 995 0
www.softengine.de • info@softengine.de