

# **kiC Dokumentenmanagement Webservice / COM-Schnittstelle**

## **Allgemeines**

Das kiC Dokumentenmanagement verfügt ab Version 41 über eine externe Schnittstelle in Form eines Webservices bzw. einer COM-Schnittstelle.

Der Webservice läuft als Dienst direkt auf dem Server, auf dem die Software installiert ist und kann im Netzwerk von beliebigen Anwendungen angesprochen werden. Der Webservice soll verwendet werden, wenn unabhängig von einem Arbeitsplatz Daten aus kiC DMS gelesen bzw. geschrieben werden müssen.

Die COM-Schnittstelle läuft auf dem Arbeitsplatz-PC des Anwenders. Die COM-Schnittstelle ist für die lokale Automatisierung gedacht ähnlich wie VBA für Microsoft Office-Anwendungen.

Die Funktionen und Signaturen sowie das Format der Parameter sind beim Webservice und bei der COM-Schnittstelle identisch. Der Unterschied liegt im Verbindungsaufbau: das COM-Object wird mittels CreateObject angelegt, der Webservice muß via TCP/IP mittels IP und Port unter Nutzung der bereitgestellten WSDL angesprochen werden.

Moderne Entwicklungsumgebungen bieten sowohl für die Zugriffe auf COM-Objekten als auch für die Anbindung von Webservices entsprechende Kapselungen.

kiC stellt für die Entwicklung ein lokales Testsystem zur Verfügung. Dieses besteht aus einer Datenbank mit den Grundstrukturen des DMS und eben dem Webservice und der COM-Schnittstelle. Im Rahmen der Bereitstellung des Testsystems werden in Abstimmung mit der Drittfirma die Strukturen in der Datenbank so festgelegt, daß diese 1:1 der zukünftigen Kundendatenbank entsprechen.

Die Webservice- und COM-Schnittstelle wird aktuell ständig erweitert, d. h. es kommen neue Funktionen hinzu. Bestehende Funktionen bleiben grundsätzlich unverändert.

## **Allgemeine Schnittstellenbeschreibung**

Daten werden im JSON-Format übergeben sofern keine expliziten Parameter für die Daten vorgesehen sind (wie z. B. username o. ä.).

In den JSON-Daten sind Datums- und Uhrzeiten immer ISO 8601 und UTC kodiert. (z.B. 2018-08-02T16:11:58.989Z). Eine Ausnahme bildet der Wert null, dieser entspricht dem Datenbankwert NULL.

Nach dem Verbindungsaufbau zur Schnittstelle muß im ersten Aufruf ein Login durchgeführt werden und vor dem Verbindungsabbau ein Logout.

Die nachfolgende Dokumentation der Funktionsaufrufe bezieht sich auf die COM-Schnittstelle. Die Schnittstelle des Webservices ist identisch nur daß als zusätzlicher Parameter beim Login ein Sessionkey zurückgegeben wird, welcher bei allen zukünftigen Aufrufen mit übergeben werden muß. Unter COM ist dies nicht notwendig, da das COM-Objekt ja direkt mit der aufrufenden Anwendung verbunden wird.

Bei erfolgreicher Ausführung eines Funktionsaufrufes ist, die zurückgegebene Zeichenkette leer, ansonsten enthält die Zeichenkette eine Fehlerbeschreibung

## COM-Schnittstelle

<b>Login</b> (const username, password: WideString): WideString		
Nach Anlegen des COM-Objekts muß das Login erfolgen. Ohne erfolgreiches Login schlagen alle weiteren Funktionsaufrufe fehl.		
Parameter	Datentyp	Beschreibung
username	WideString	Als Benutzername muß der Name eines im kiC DMS angelegten Benutzers übergeben werden. Alle weiteren Funktionen arbeiten im Rechtesystem des Benutzers, d. h. wenn über die COM-Schnittstelle Dokumente angelegt werden sollen dann muß der Benutzer im Rahmen der zugewiesenen Rollen und Berechtigungen über das Recht zum Anlegen von Dokumenten verfügen. Wenn nicht, liefern die aufgerufenen Funktionen Fehlermeldungen zurück.
password	WideString	Kennwort des Benutzers

<b>Logout</b> : WideString		
Vor dem Zerstören des COM-Objekts muß ein Logout erfolgen. Das anschließende Zerstören des COM-Objekts erfolgt in der Regel durch Setzen der Variable, welche das COM-Objekt hält, auf NIL oder NULL.		
Parameter	Datentyp	Beschreibung
keine		

<b>GetRolesAndClients</b> (out data: WideString): WideString		
Nach erfolgreichem Login sollten mit dieser Funktion die dem angemeldeten Benutzer zugeordneten Rollen und Mandanten abgefragt werden. Beim Anlegen eines Objekts muß immer eine Rolle und ein Mandant übergeben werden (Besitzerrolle / -mandant des Objekts repräsentiert durch die IDs, welche mit dieser Funktion ermittelt werden können).		
Parameter	Datentyp	Beschreibung
data	JSON WideString	Im folgenden Beispiel ist der der angemeldete Benutzer in der "Rolle 1" mit ID=1 und in der "Rolle 2" mit ID=6. Den Rollen sind jeweils die Mandanten "Mandant x" mit den jeweiligen IDs zugeordnet - "Rolle 1" also die Mandanten 1-4 und "Rolle 2" die Mandanten 1-3.  <pre>{   "roles": [     {       "id": 1,       "name": "Rolle 1",       "clients": [</pre>

		<pre>         {           "id": 2,           "name": "Mandant 1"         },         {           "id": 3,           "name": "Mandant 2"         },         {           "id": 4,           "name": "Mandant 3"         },         {           "id": 5,           "name": "Mandant 4"         }       ],       {         "id": 6,         "name": "Rolle 2",         "clients": [           {             "id": 7,             "name": "Mandant 1"           },           {             "id": 8,             "name": "Mandant 2"           },           {             "id": 9,             "name": "Mandant 3"           }         ]       }     ]   } </pre>
--	--	--

<b>GetCategories(parentId: Integer; recursive: WordBool; out data: WideString): WideString</b>		
Lädt alle für den Benutzer sichtbare Kontaktkategorien.		
Parameter	Datentyp	Beschreibung
parentId	Integer	Die ID der Kategorie von der die Unterkategorien gelesen werden sollen. 0 liefert alle Kategorien zurück sofern recursive=true ist
recursive	Boolean	Wenn "true" dann werden ausgehend von der Kategorie mit parentId alle Unterkategorien gelesen. Bei "false" nur die Ebene unter der übergebenen Kategorie.
data	JSON WideString	Liefert die Kategorien gemäß Einstellung der beiden anderen Parameter zurück. Im Beispiel existiert auf der Wurzel eine Kategorie 1, 11 und 2. Kategorie 1 hat jeweils eine Unterkategorie mit einer weiteren Unterkategorie, Kategorie 11 hat keine Unterkategorien und Kategorie 2 hat eine Unterkategorie.  <pre> {   "categories": [ </pre>

		<pre> {   "id": 1,   "name": "Kategorie 1",   "children": [     {       "id": 2,       "name": "Kategorie 10",       "children": [         {           "id": 3,           "name": "Kategorie 100",           "children": []         }       ]     }   ],   "id": 4,   "name": "Kategorie 11",   "children": [] }, {   "id": 5,   "name": "Kategorie 2",   "children": [     {       "id": 6,       "name": "Kategorie 20",       "children": []     }   ] } ] </pre>
--	--	--

<b>GetContacts(categoryId: Integer; out data: WideString): WideString</b>		
Lädt alle für den Benutzer sichtbaren Kontakte in der übergebenen Kategorie.		
Parameter	Datentyp	Beschreibung
categoryID	Integer	ID der Kategorie, deren Kontakte zurückgeliefert werden sollen. 0 liefert alle für den Benutzer sichtbaren Kontakte.
data	JSON WideString	<p>Liste der Kontakte. Zur Beschreibung der Kontaktfelder siehe die Funktion GetContact.</p> <pre> {   "contacts": [     {       "id": 191,       "name": "name1\"{} [] \\\"üöäßÜÖÄ&lt;\r\n&gt;",       "name1": "name11",       "name2": "",       "type": "person",       "title": "title1",       "dob": "2018-07-17T09:12:13.000Z",       "functionname": "functionname1",       "functionnameshort": "1",       "salutationtype": "Frau",       "addresses": [         {           "id": 196, </pre>

		<pre> "city": "city2", "country": "country2", "district": "district2", "email1": "email12", "email2": "email22", "fax1": "fax12", "fax2": "fax22", "mobile1": "mobile12", "mobile2": "mobile22", "name": "name2", "phone1": "phone12", "phone2": "phone22", "poBox": "pobox2", "poboxcity": "poboxcity2", "poboxzip": "2", "street": "street2", "www": "www2", "zip": "zip2", "persons": [] }, {   "id": 197,   "city": "city3",   "country": "country3",   "district": "district3",   "email1": "email13",   "email2": "email23",   "fax1": "fax13",   "fax2": "fax23",   "mobile1": "mobile13",   "mobile2": "mobile23",   "name": "name3",   "phone1": "phone13",   "phone2": "phone23",   "poBox": "pobox3",   "poboxcity": "poboxcity3",   "poboxzip": "3",   "street": "street3",   "www": "www3",   "zip": "zip3",   "persons": [] } ], "categories": [   {     "id": 189,     "name": "cat0"   } ] }, {   "id": 199,   "name": "name4",   "name1": "name14",   "name2": "name24",   "type": "company",   "title": "",   "dob": null,   "functionname": "functionname4",   "functionnameshort": "4",   "salutationtype": "Firma",   "addresses": [     {       "id": 204,       "city": "city5",       "country": "country5",       "district": "district5",       "email1": "email15", </pre>
--	--	---

		<pre> "email2": "email25", "fax1": "fax15", "fax2": "fax25", "mobile1": "mobile15", "mobile2": "mobile25", "name": "name5", "phone1": "phone15", "phone2": "phone25", "poBox": "pobox5", "poboxcity": "poboxcity5", "poboxzip": "5", "street": "street5", "www": "www5", "zip": "zip5", "persons": [] }, {   "id": 205,   "city": "city6",   "country": "country6",   "district": "district6",   "email1": "email16",   "email2": "email26",   "fax1": "fax16",   "fax2": "fax26",   "mobile1": "mobile16",   "mobile2": "mobile26",   "name": "name6",   "phone1": "phone16",   "phone2": "phone26",   "poBox": "pobox6",   "poboxcity": "poboxcity6",   "poboxzip": "6",   "street": "street6",   "www": "www6",   "zip": "zip6",   "persons": [     {       "id": 206,       "department": "department7",       "email": "email7",       "fax": "fax7",       "functionname": "functionname7",       "functionnameshort": "7",       "mobile": "mobile7",       "phone": "phone7",       "roomnumber": "roomnumber7",       "contactid": 191,       "contactsurname": "name1\"{} []\\\"üöäßÜÖÄ&lt;r\\n&gt;",       "contactforename": "name11"     }   ] } ], "categories": [   {     "id": 189,     "name": "cat0"   } ] } ] } </pre>
--	--	---

<b>GetContact</b> (contactId: Integer; out data: WideString): WideString		
Lädt die Daten eines Kontakts.		
Parameter	Datentyp	Beschreibung
contactId	Integer	ID des zu ladenden Kontakts.
data	JSON WideString	<p>Daten des Kontakts.</p> <p>Kontaktyp kann "person" oder "company" sein.</p> <p>Jeder Kontakt hat 1..n Adressen.</p> <p>Ein Firmenkontakt zusätzlich noch pro Adresse 0..n Ansprechpartner.</p> <p>Ein Kontakt kann 0..n Kategorien zugeordnet sein.</p> <pre> {   "contact": {     "id": 23,     "name": "name1\{"} [] \\"üöäßÜÖÄ&lt;\r\n&gt;",     "name1": "name11",     "name2": "",     "type": "person",     "title": "title1",     "dob": "2018-07-17T09:12:13.000Z",     "functionname": "functionname1",     "functionnameshort": "1",     "salutationtype": "Frau",     "addresses": [       {         "id": 0,         "city": "city2",         "country": "country2",         "district": "district2",         "email1": "email12",         "email2": "email22",         "fax1": "fax12",         "fax2": "fax22",         "mobile1": "mobile12",         "mobile2": "mobile22",         "name": "name2",         "phone1": "phone12",         "phone2": "phone22",         "poBox": "pobox2",         "poboxcity": "poboxcity2",         "poboxzip": "2",         "street": "street2",         "www": "www2",         "zip": "zip2",         "persons": []       },       {         "id": 0,         "city": "city3",         "country": "country3",         "district": "district3",         "email1": "email13",         "email2": "email23",         "fax1": "fax13",         "fax2": "fax23",         "mobile1": "mobile13",         "mobile2": "mobile23",         "name": "name3",         "phone1": "phone13",         "phone2": "phone23",       }     ]   } } </pre>



		<pre>                 "poBox": "pobox3",                 "poboxcity": "poboxcity3",                 "poboxzip": "3",                 "street": "street3",                 "www": "www3",                 "zip": "zip3",                 "persons": []             }         ],         "categories": []     } } </pre>
--	--	--

<b>GetDocument</b> (documentId: Integer; out data: WideString): WideString		
Lädt die Metadaten eines Dokuments.		
Parameter	Datentyp	Beschreibung
documentId	Integer	ID des zu ladenden Dokuments.
data	JSON WideString	Metadaten des Dokuments.  <pre> {   "document": {     "id": 12,     "name": "name1\("{} []\\"üöäßÜÖÄ&lt;",     "description": "description1\r\ndescription1",     "folderid": 13,     "attributes": [       {         "connector": "connector1",         "value": "99"       },       {         "connector": "connector2",         "value": "2018-05-07"       },       {         "connector": "connector3",         "value": "true"       },       {         "connector": "connector4",         "value": "string1"       },       {         "connector": "connector5",         "value": "2018-07-23T13:16:17.000Z"       },       {         "connector": "connector6",         "value": "2018-07-22T18:16:17.000Z"       },       {         "connector": "connector7",         "value": "1.9"       },       {         "connector": "connector8",         "value": "memo1"       }     ]   } } </pre>

<b>GetDocumentContent</b> (documentId: Integer; out content: OleVariant): WideString		
Lädt die binären Daten eines Dokuments (Word, PDF, usw.).		
Parameter	Datentyp	Beschreibung
documentId	Integer	ID des zu ladenden Dokuments.
contact	OleVariant vom Typ SafeArray	Binäre Daten des Dokuments.

<b>CreateDocument</b> (const data: WideString; content: OleVariant; roleId: Integer; clientId: Integer; out documentId: Integer): WideString		
<p>Legt ein neues Dokument an. Die Metadaten des Dokuments müssen in data übergeben werden, die binären Daten in content. roleID und clientID muß eine gültige Kombination aus Rolle/Mandant sein (Besitzerrolle/Mandant des Dokuments). Die Werte hierzu müssen nach dem Login mit GetRolesAndClients ermittelt werden. documentId liefert als Rückgabe die ID des neu angelegten Dokuments.</p>		
Parameter	Datentyp	Beschreibung
data	JSON WideString	<p>JSON mit den Metadaten des neu anzulegenden Dokuments.</p> <p>Name und folderid sind Pflichtfelder.</p> <pre>{   "document": {     "name": "name1\{" } [] \\\"üöäßÜÖÄ&lt;",     "description": "description1\r\ndescription1",     "folderid": 12,     "attributes": [       {         "connector": "connector1",         "value": "99"       },       {         "connector": "connector2",         "value": "2018-05-07"       },       {         "connector": "connector3",         "value": "true"       },       {         "connector": "connector4",         "value": "string1"       },       {         "connector": "connector5",         "value": "2018-07-23T13:16:17.000Z"       },       {         "connector": "connector6",         "value": "2018-07-22T18:16:17.000Z"       },       {         "connector": "connector7",</pre>

		<pre>         "value": "1.9"       },       {         "connector": "connector8",         "value": "memo1"       }     ]   } } </pre>
content	OleVariant vom Typ SafeArray	Binäre Daten
roleId	Integer	ID der Besitzerrolle
clientId	Integer	ID des Besitzermandanten
documentId	Integer	ID des neu angelegten Dokuments

<b>DeleteDocument</b> (documentId: Integer): WideString		
Löscht ein Dokument.		
Parameter	Datentyp	Beschreibung
documentId	Integer	ID des zu löschenden Dokuments.

<b>UpdateDocument</b> (documentId: Integer; const data: WideString; content: OleVariant): WideString		
Aktualisiert ein Dokument. Es werden nur die Metadaten aktualisiert, welche in data übergeben werden. Alle anderen Metadaten bleiben unverändert. Sofern content leer ist, werden die binären Daten nicht verändert.		
Parameter	Datentyp	Beschreibung
documentId	Integer	ID des zu aktualisierenden Dokuments.
data	JSON WideString	Metadaten des Dokuments. <pre> {   "document": {     "name": "name1\ "{} [] \ \ \"üöäßÜÖÄ&lt;",     "description": "description1\r\ndescription1",     "folderid": 12,     "attributes": [       {         "connector": "connector1",         "value": "99"       },       {         "connector": "connector2",         "value": "2018-05-07"       },       {         "connector": "connector3",         "value": "true"       },       {         "connector": "connector4",         "value": "string1"       },       {         "connector": "connector5", </pre>

		<pre>         "value": "2018-07-23T13:16:17.000Z"       },       "connector": "connector6",       "value": "2018-07-22T18:16:17.000Z"     },     "connector": "connector7",     "value": "1.9"   },   "connector": "connector8",   "value": "memo1" ] } </pre>
content	OleVariant vom Typ SafeArray	Binäre Daten des Dokuments.

## Webservice

Wie zu Anfang geschrieben sind die Funktionen des Webservices identisch zu denen des COM-Objekts mit Ausnahme der SessionID.

Diese wird beim Login an den Aufrufer zurückgegeben und muß dann bei jedem weiteren Funktionsaufruf mit übergeben werden.

Beispiel:

```
function Login(const aUserName, aPassword: WideString; out aSessionID: WideString):  
Widestring;
```

```
function Logout(const aSessionID: WideString): WideString;
```

```
function GetRolesAndClients(const aSessionID: WideString; out data: WideString): WideString;
```

```
function GetCategories(const aSessionID: WideString; aParentId: Integer; aRecursive: Boolean; var  
data: WideString): WideString;
```

usw.

Die SessionID ist also beim Webservice immer der erste Parameter, danach erfolgen die gleichen Parameter wie bei der entsprechenden COM-Funktion.