**Functies die worden aangeboden via webservices (soap)**

De CorsaKoppeling heeft de volgende koppelingen die worden aangeboden aan de webformulieren (Kodision).

Heel belangrijk in de werking is dat de werking stateless moet zijn, dit betekend dat Hein zeker weet wanneer iets is opgeslagen: dan krijgt hij een corsa-registratie-nummer terug en weet hij dat alles goed is verlopen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Functienaam:** | **Wat doet het:** | **Voorbeeld parameters:** | **Voorbeeld returnwaarde:** | **Bij fouten:** |
| string Version(string version); | Geeft huidige versie terug van de koppeling en de corsa-wrapper |  | 2.0-20181116045205 | Werking is stateless, dus bij fouten altijd exceptie |
| string Info(); | Geeft de bouwdatum terug van de koppeling en verdere versie informatie terug | version: 2.0-20181116045205 | debug build: 'False' userdomainname: 'SWFFB01' server: 'SWFFB01' enviroment username: 'Corsa\_user' principal username: 'S-1-5-21-3490545410-705332683-3954547330-1011' userprincipal sid: 'S-1-5-21-3490545410-705332683-3954547330-1011' osversion: 'Microsoft Windows NT 6.3.9600.0' is64bits: 'False' directory: 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files\root\06f91b6d\8ff6aa25\assembly\dl3\4ba8078e\07f5b652\_c47dd401\CorsaKoppeling.dll'  read registry corsaversion: '7.2.151.178' | Werking is stateless, dus bij fouten altijd exceptie |
| string opslaanInkNatuurlijkPersoon(string afzenderbsn, string vertrouwelijkheid, string behandelaarnaam, string onderwerp, string inhoud, string filename, byte[] filedata); | Maak een document van type "InkomendDocument" (INK) en geeft een documentnummer terug (corsa-stuknummer) de registratie wordt meteen gekoppeld aan de betreffende persoon in Corsa kanaal = "Internet" | afzenderbsn: 598596823 vertrouwelijkheid: V-int onderwerp: TestOnderwerp inhoud: TestLangeTekst filename: hello.txt fildata: aGVsbG8= (base64) | I18.700453 | Werking is stateless, dus bij fouten altijd exceptie |
| string opslaanInkNietNatuurlijkPersoon(string afzenderkvk, string vertrouwelijkheid, string behandelaarnaam, string onderwerp, string inhoud, string filename, byte[] filedata); | Maak een document van type "InkomendDocument" (INK) en geeft een documentnummer terug (corsa-stuknummer) de registratie wordt meteen gekoppeld aan de betreffende bedrijf in Corsa | afzenderkvk: 33277255 vertrouwelijkheid: V-int onderwerp: TestOnderwerp inhoud: TestLangeTekst filename: hello.txt fildata: aGVsbG8= (base64) | I18.700453 | Werking is stateless, dus bij fouten altijd exceptie |
| string opslaanBijlage(string registratienummer, string filename, byte[] filedata); | Maak een document van type "Bijlage" (BIJL) en geeft een documentnummer terug (corsa-stuknummer) de registratie wordt meteen gekoppeld aan het meegestuurde document | registratienummer: I18.700453 filename: hello2.txt filedata: aGVsbG8y (base64) | BIJL18.700453 | Werking is stateless, dus bij fouten altijd exceptie |
| string opslaanBijlageDocumenttype(string registratienummer, string documenttype, string filename, byte[] filedata); | Maak een document van type "Bijlage" (BIJL) en geeft een documentnummer terug (corsa-stuknummer) de registratie wordt meteen gekoppeld aan het meegestuurde document | registratienummer: I18.700453 documenttype: Bijlage filename: hello2.txt filedata: aGVsbG8y (base64) | BIJL18.700453 | Werking is stateless, dus bij fouten altijd exceptie |

Configuratie instellingen:

* corsa\_database: ds\_test\_1900
* corsa\_user:     WebFor1
* corsa\_password: \*\*\*\*\*\*\*
* corsa\_verbose   true

**Functie opslaanInkNatuurlijkPersoon**

    public string opslaanInkNatuurlijkPersoon(string afzenderbsn, string vertrouwelijkheid, string behandelaarnaam, string onderwerp, string inhoud, string filename, byte[] filedata)

    {

        var corsa = new CorsaKoppeling.Corsa(Convert.ToBoolean(ConfigurationManager.AppSettings["corsa\_verbose"]));

        corsa.Connect(ConfigurationManager.AppSettings["corsa\_database"], ConfigurationManager.AppSettings["corsa\_user"], ConfigurationManager.AppSettings["corsa\_password"]);

        var identificatie = corsa.getBsnIdentificatie(afzenderbsn) ?? corsa.createBsnIdentificatie(afzenderbsn);

        var velden = new CorsaKoppeling.PoststukVelden(vertrouwelijkheid: vertrouwelijkheid, onderwerp: onderwerp, bericht: inhoud, behandelaarnaam: behandelaarnaam);

        velden.AddBsnIdentificatie(identificatie);

        var kenmerken = new CorsaKoppeling.PoststukKenmerken();

        var registratieIdentificatie = corsa.CreateMetaDocument("INK", velden, kenmerken);

        corsa.AddDocument("INK", registratieIdentificatie, filename, filedata);

        corsa.Disconnect();

        // HACK HACK, gebruik dezelfde vertrouwelijkheid als de aanmaker

        GebruikteVertrouwelijkheden.Add(registratieIdentificatie, vertrouwelijkheid);

        return registratieIdentificatie;

    }

**Functie opslaanInkNietNatuurlijkPersoon**

        var corsa = new CorsaKoppeling.Corsa(Convert.ToBoolean(ConfigurationManager.AppSettings["corsa\_verbose"]));

        corsa.Connect(ConfigurationManager.AppSettings["corsa\_database"], ConfigurationManager.AppSettings["corsa\_user"], ConfigurationManager.AppSettings["corsa\_password"]);

        var identificatie = corsa.getKvkIdentificatie(afzenderkvk) ?? corsa.createKvkIdentificatie(afzenderkvk);

        var velden = new CorsaKoppeling.PoststukVelden(vertrouwelijkheid: vertrouwelijkheid, onderwerp: onderwerp, bericht: inhoud, behandelaarnaam: behandelaarnaam);

        velden.AddKvkIdentificatie(identificatie);

        var kenmerken = new CorsaKoppeling.PoststukKenmerken();

        var registratieIdentificatie = corsa.CreateMetaDocument("INK", velden, kenmerken);

        corsa.AddDocument("INK", registratieIdentificatie, filename, filedata);

        corsa.Disconnect();

        // HACK HACK, gebruik dezelfde vertrouwelijkheid als de aanmaker

        GebruikteVertrouwelijkheden.Add(registratieIdentificatie, vertrouwelijkheid);

        return registratieIdentificatie;

**Functie opslaanBijlage**

        var corsa = new CorsaKoppeling.Corsa(Convert.ToBoolean(ConfigurationManager.AppSettings["corsa\_verbose"]));

        corsa.Connect(ConfigurationManager.AppSettings["corsa\_database"], ConfigurationManager.AppSettings["corsa\_user"], ConfigurationManager.AppSettings["corsa\_password"]);

        // HACK HACK, gebruik dezelfde vertrouwelijkheid als de aanmaker

        var vertrouwelijkheid = GebruikteVertrouwelijkheden[registratienummer];

        // maak de registratie aan

        var velden = new CorsaKoppeling.PoststukVelden(vertrouwelijkheid);

        var kenmerken = new CorsaKoppeling.PoststukKenmerken();

        var bijlagenummer = corsa.CreateMetaDocument("BIJL", velden, kenmerken);

        // stop het document erin

        corsa.AddDocument("BIJL", bijlagenummer, filename, filedata);

        // koppel aan het hoofdocument

        corsa.LinkDocuments(registratienummer, bijlagenummer);

        corsa.Disconnect();

        return bijlagenummer;

**Functie opslaanBijlageDocumenttype**

        var corsa = new CorsaKoppeling.Corsa(Convert.ToBoolean(ConfigurationManager.AppSettings["corsa\_verbose"]));

        corsa.Connect(ConfigurationManager.AppSettings["corsa\_database"], ConfigurationManager.AppSettings["corsa\_user"], ConfigurationManager.AppSettings["corsa\_password"]);

        // HACK HACK, gebruik dezelfde vertrouwelijkheid als de aanmaker

        var vertrouwelijkheid = GebruikteVertrouwelijkheden[registratienummer];

        // maak de registratie aan

        var velden = new CorsaKoppeling.PoststukVelden(vertrouwelijkheid);

        var kenmerken = new CorsaKoppeling.PoststukKenmerken();

        kenmerken.Add("DOCTYP", documenttype);

        var bijlagenummer = corsa.CreateMetaDocument("BIJL", velden, kenmerken);

        // stop het document erin

        corsa.AddDocument("BIJL", bijlagenummer, filename, filedata);

        // koppel aan het hoofdocument

        corsa.LinkDocuments(registratienummer, bijlagenummer);

        corsa.Disconnect();

        return bijlagenummer;

**TDSAPI.dll via de CorsaKoppeling.Corsa**

public void Connect(string database, string username, string password)

        {

            acorsa = new TDSAPI.ASCORSA();

            acorsa.ShowErrorMsg = verbose;

            acorsa.DataBaseID = database;

            if (acorsa.Connect(username, password, false) == false) throw new CorsaException("acorsa.Connect('" + database + "', '" + username + "', '\*\*\*\*\*\*\*\*')", acorsa);

            documents = new TDSAPI.DSDOCUMENTS();

            documents.ShowErrorMsg = verbose;

            documents.DataBaseID = database;

            if (documents.Connect(username, password, false) == false) throw new CorsaException("documents.Connect('" + database + "', '" + username + "', '\*\*\*\*\*\*\*\*')", acorsa);

        }

        public void Disconnect()

        {

            // expliciet! ivm. stomme dcom mem-management

            documents.Disconnect();

            System.Runtime.InteropServices.Marshal.ReleaseComObject(documents);

            documents = null;

            System.Runtime.InteropServices.Marshal.ReleaseComObject(acorsa);

            acorsa = null;

        }

        private void AddDocument(string documenttypecode, string corsaobjectid, string filename, string description, System.IO.FileInfo file)

        {

            // TDSAPI.IDSDOCUMENT doc = documents.Add();

            var doc = documents.Add();

            doc.ObjectType = "S";

            doc.ObjectID = corsaobjectid;

            doc.ObjectSoort = documenttypecode;

            doc.Description = description;

            doc.Name = filename;

            doc.Version = "0"; //  Automatically detect correct version number

            if (doc.Store(file.FullName, 7) == false) throw new CorsaException("document.Store('" + file.FullName + "', 7)", acorsa);

            doc.Close();

            // expliciet! ivm. stomme dcom mem-management

            System.Runtime.InteropServices.Marshal.ReleaseComObject(doc);

            doc = null;

        }

        public void AddDocument(string documenttypecode, string corsaobjectid, string filename, byte[] filedata)

        {

            var fi = new System.IO.FileInfo(System.IO.Path.GetTempPath() + DateTime.Now.ToString("yyyyMMddhhmmss") + "-" + filename);

            var writer = new System.IO.BinaryWriter(fi.OpenWrite());

            writer.Write(filedata);

            writer.Flush();

            writer.Close();

            AddDocument(documenttypecode, corsaobjectid, filename, filename, fi);

            if (!verbose) { fi.Delete();}

            else { System.Diagnostics.Debug.WriteLine("Stored the file:" + filename + ", keeping the filedata in : " + fi.FullName);  }

        }

        public void LinkDocuments(string documentnummer, string bijlagenummer)

        {

            string omschrijving = "bijlage: " + bijlagenummer + " bij document: " + documentnummer;

            if (acorsa.ModObjectRelations(1 /\* create relation\*/, "S", documentnummer, "S", bijlagenummer, "BIJ") == false)  throw new CorsaException("acorsa.ModObjectRelations(1, 'S', '" + documentnummer + "', 'S', '" + bijlagenummer + "' , 'BIJ')", acorsa);

        }

**Velden voor TDSAPI.dll**

          var now = DateTime.Now.ToString("dd/MM/yyyy");

Add("poststuk.dat\_afh\_str", now);

            Add("poststuk.onderwerp", onderwerp);

            Add("poststuk.inhoud1", bericht);

            Add("poststuk.v\_plaats\_id", behandelaarnaam);

            Add("poststuk.reg\_datum", now);         // DOET HET NIET!

            Add("poststuk.dat\_poststuk", now);      // DOET HET NIET!

            Add("poststuk.kenmerk", onderwerp); // DOET HET NIET!

            Add("obj\_vert.vertrouw\_id", vertrouwelijkheid);

      Natuurlijk persoon:

            Add("poststuk.soort\_ext", "P");

            Add("poststuk.relatie\_id", aanmelderIdentificatie);

                Bedrijf:

Add("poststuk.soort\_ext", "I");

            Add("poststuk.relatie\_id", aanmelderIdentificatie);

**Kenmerken voor TDSAPI.dll**

            var now = DateTime.Now.ToString("dd-MM-yyyy");

            Add("ontvdat", now);

            Add("kanaal", "Internet");

**Koppelen personen:**

        public string getBsnIdentificatie(string bsnnummer)

        {

            var query = "1" + (char)2 + "K" + (char)2 + "BRSpSofi" + (char)2 + "AND" + (char)2 + "=" + (char)2 + bsnnummer;

            var spnummer = acorsa.QueryExecute("P", query);

            if (spnummer == "") return null;

            return spnummer;

        }

        public string createBsnIdentificatie(string bsnnummer)

        {

            string ObjId = "";       // Object id to update

            string ObjSrt = "1";      // Object kind

            var velden = new CorsaParameterList();

            var kenmerken = new CorsaParameterList();

            kenmerken.Add("BRSpSofi", bsnnummer);

            string ErrorMsg = "";     // Output If an error occurs then this parameter contains the message

            var result = acorsa.CreateMetaPerson(ObjId, ObjSrt, velden.Keys(), velden.Values(), kenmerken.Keys(), kenmerken.Values(), ref ErrorMsg);

            if (ErrorMsg != null) throw new CorsaException(ErrorMsg);

            return result;

        }

        public string getKvkIdentificatie(string afzenderkvk)

        {

            var query = "1" + (char)2 + "K" + (char)2 + "BRSKVK" + (char)2 + "AND" + (char)2 + "=" + (char)2 + afzenderkvk;

            var spnummer = acorsa.QueryExecute("E", query);

            if (spnummer == "") return null;

            return spnummer;

        }

        public string createKvkIdentificatie(string afzenderkvk)

        {

            string ObjId = "";       // Object id to update

            string ObjSrt = "1";      // Object kind

            var velden = new CorsaParameterList();

            var kenmerken = new CorsaParameterList();

            kenmerken.Add("BRSKVK", afzenderkvk);

            string ErrorMsg = "";     // Output If an error occurs then this parameter contains the message

            var result = acorsa.CreateMetaOrganization(ObjId, ObjSrt, velden.Keys(), velden.Values(), kenmerken.Keys(), kenmerken.Values(), ref ErrorMsg);

            if (ErrorMsg != null) throw new CorsaException(ErrorMsg);

            return result;

        }