



Dokumentation: Acubiz EMS SOAP API

Version 1.7 / 28-08-2017

Indholdsfortegnelse

Adgang til Acubiz SOAP API	3
Arkitektur.....	3
Adgang via .NET.....	3
Generelle principper i Acubiz SOAP API.....	5
Tilgængelige funktioner	6
Generelle metoder.....	6
Metoder og klasser til håndtering af brugere.....	6
Metoder og klasser til håndtering af EMS databaser.....	7
Metoder og klasser til håndtering af dimensioner.....	7
Metoder og klasser til håndtering af eksport data.....	8
Metoder og klasser til håndtering af IMPORT data.....	9
Brugere: Læsning, opdatering og oprettelse	10
Listning af brugere.....	10
Oprettelse af brugere.....	11
Opdatering af brugere.....	12
Dimensioner: Læsning, opdatering og oprettelse	13
Opslag efter EMS databaser.....	13
Listing af dimensioner.....	14
Oprettelse af dimensioner.....	16
Opdatering af dimensioner.....	17
Import af data	18
ImportCSVdata.....	18
ImportXMLdata.....	18
Eksport af data	19
EXPORTCSVDATA.....	19
EXPORTXMLDATA.....	20
Index	22

ADGANG TIL ACUBIZ SOAP API

Forskellige sprog og frameworks har deres egen funktionalitet for at benytte en SOAP-baseret web service. De fleste af disse involverer automatisk generering klient kode baseret på WSDL, men det er også muligt at bygge XML envelopes helt manuelt.

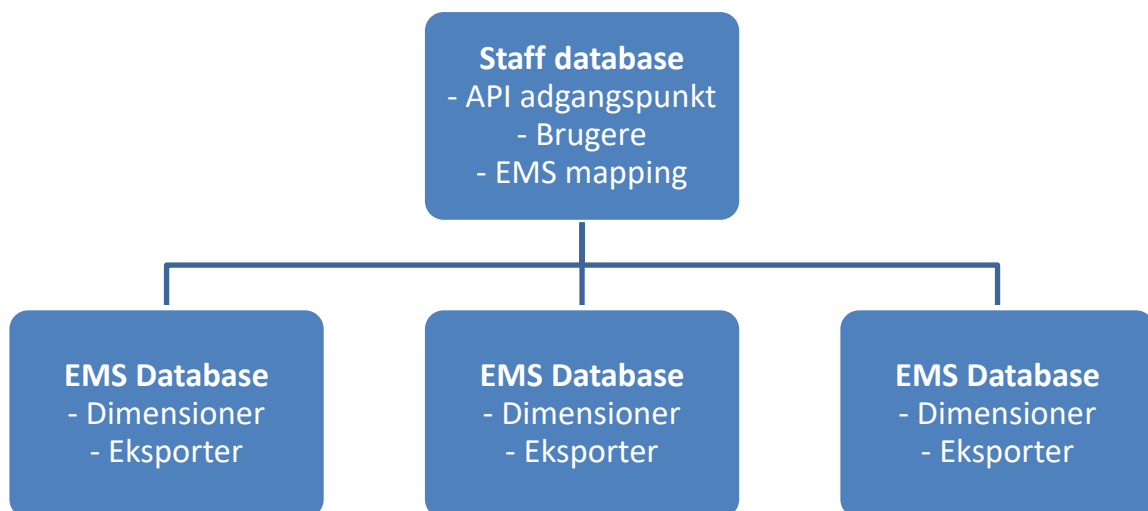
Du skal sikre, at session cookie bliver sendt med i alle transaktioner. Nogle framesets vil gøre dette automatisk, men ikke alle.

ARKITEKTUR

Acubiz EMS er opbygget med en Staff og konfigurationsdatabase, der er det centrale i Acubiz EMS. Derudover er der en eller flere EMS databaser, afhængig af hvor mange segmenteringer man ønsker.

Kontaktpunktet for Acubiz SOAP API er Staff databasen, som indeholder brugere og kortlægning af de tilknyttede EMS databaser. Igennem API'et kan man også tilgå hver enkelt EMS databases dimensioner og eksporter.

Denne opbygning gør, at man skal hente en handler for den EMS database man skal tilgå, for at kunne håndtere dimensioner og eksporter.



ADGANG VIA .NET

Via Service References menuen i Visual Studio, kan du tilføje en service reference til Acubiz SOAP API.

URL'en til dit SOAP API er angivet i API modulets indstillinger.

For at kunne læse WSDL beskrivelsen, skal du autentikere med en bruger på dit system. Det anbefales at du benytter en bruger der specifikt er oprettet til API eller en admin bruger (se Autentikering).

Når du har gjort det, skal du modificere service referencen til at bruge basic authentication og til at kunne sende og modtage større pakker. Det gør du ved at ændre service reference bindingen til at indeholde en `clientCredentialType` der hedder "Basic":

Altså: "

```
<security mode="Transport">  
  <transport clientCredentialType="Basic" />  
</security>
```

Den samlede `<bindings>` vil således se sådan ud. Da du importerede Webservice, blev denne genereret automatisk, så derfor kan du ikke indsætte denne som den er, men skal flette dit `<security>` tag ind.

```
<bindings>  
  <basicHttpBinding>  
    <binding name="DominoSoapBinding" maxBufferSize="999999" maxReceivedMessageSize="999999">  
      <security mode="Transport">  
        <transport clientCredentialType="Basic" />  
      </security>  
    </binding>  
  </basicHttpBinding>  
</bindings>
```

For at autentikere og tilslutte til webseriven, benyttes følgende fremgangsmåde. Vi har her kaldt vores service reference for Acubiz.

```
Acubiz.SessionClient WS = new Acubiz.SessionClient();  
WS.ClientCredentials.UserName.UserName = <brugernavn>  
WS.ClientCredentials.UserName.Password = <adgangskode>  
WS.Open();
```

For at læse hvilken state som service referencen er i, kan der benyttes følgende kald:
`WS.State.ToString();`

Det vil give tre svar:

- Created
- Opened
- Closed

Authentikering

En hvilken som helst bruger, der er oprettet i systemet, kan autentikere op imod webservicen for at hente WSDL og for at kommunikere med Acubiz.

Det er dog kun muligt at benytte funktioner og se data, relateret til brugerens adgang.

Det vil altså sige, at hvis man ikke er admin på systemet, kan man ikke oprette eller opdatere brugere og dimensioner, ligesom det kun er finansbrugere der kan eksportere.

GENERELLE PRINCIPPER I ACUBIZ SOAP API

Klassen `SessionClient` indeholder metoder, der benyttes til at læse, skrive og opdatere data i Acubiz EMS.

Data bæres i tilhørende klasser, så som USER, EMS og DIMENSION. Disse klasser modtages med data og USER og DIMENSION klasserne kan sendes retur med opdateringer. Når der sendes en opdatering, returneres det tilrettede objekt med eventuelle fejlkoder.

Metodekald hvor der returneres array af data

I metodekald, hvor der returneres en klasse med ARRAYOFUSER, ARRAYOFEMS eller ARRAYOFDIMENSION, kan man angive hvor mange man vil have retur samt hvor i den samlede liste man gerne vil starte.

Denne mulighed gives for at kunne styre hvor store datamængder der sendes retur. Her kan man kunne angive integer værdien 0 i begge tilfælde, som returnerer alle elementer – eller i tilfælde af større lister, en delmængde.

I klassen ARRAYOFUSER bliver der returneret hvor mange elementer der er i arrayet (COUNT) og hvorfra i den samlede liste første element kommer (POSITION). For at få flere data, kan man foretage kaldet igen, med angivelse af position til POSITION + COUNT.

1.1.1.1 Eksempel

Ønsker man at hente alle dimensioner, vil man starte med at få fat i den EMS database man vil søge i, og efterfølgende hente dimensionerne med metoden GETDIMENSIONS(EMS, int POSITION, int COUNT);

I dette kald henter vi referencen til EMS databasen "emsid" og efterfølgende henter de første 1000 elementer.

```
Acubiz.EMS ems = WS.GETEMS("emsid");  
Acubiz.ARRAYOFDIMENSION dimensions = WS.GETDIMENSIONS(ems, 1, 1000);
```

Ønsker man at hente alle dimensioner, kan man ændre kaldet til:

```
Acubiz.EMS ems = WS.GETEMS("emsid");  
Acubiz.ARRAYOFDIMENSION dimensions = WS.GETDIMENSIONS(ems, 0, 0);
```

TILGÆNGELIGE FUNKTIONER

Klassen `SessionClient` indeholder følgende API funktioner, der benyttes til at læse, skrive og opdatere data i Acubiz EMS.

GENERELLE METODER

Disse metoder returnerer information om den bruger der er logget ind med, hvilken version af API'et der benyttes samt funktioner til håndtering og beskrivelse af fejl.

- `APIVERSION()`
- `GETCURRENTUSER()`
- `GETUSERNAME()`
- `GETEFFECTIVEUSERNAME()`
- `ERRORSTRING()`
- `ISERROR()`

METODER OG KLASSER TIL HÅNDTERING AF BRUGERE

Disse funktioner benyttes til at finde en eller flere brugere, oprette en bruger eller opdatere en bruger.

Informationer om brugeren bæres i en `USER` klasse, der returneres når man slår en bruger op, opretter en bruger eller opdatere en bruger.

Skal man modificere en bruger, slår man først brugeren op og benytter derefter det `USER` objekt der returneres til at opdatere de data man ønsker. Til sidst returneres `USER` objektet og et opdateret `USER` objektet modtages retur, med eventuelle fejlkoder.

Skal man hente flere brugere, returneres en klasse `ARRAYOFUSER`, der ud over et array med `USER` objekter også indeholder information om hvor mange der returneres og hvorfra i den samlede liste.

Metoder

- `FINDUSER(string)`
- `GETUSERS(int, int)`
- `CREATEUSER(string, string, string, string, string, bool, bool, bool)`
- `UPDATEUSER(Acubiz.USER)`

Klasser

- `USER`
 - `FirstName : String`
 - `LastName : String`
 - `Initials : String`
 - `Email : String`
 - `EmployeeID : String`
 - `DefaultEMS : EMS`
 - `AcessableEMS : arrayOfEMS`
 - `Authoriser : User`
 - `AltAuthoriser : User`
 - `Secretary : User`
 - `Booker : User`
 - `ApproverLimit : Double`

- CashAccount : String
 - SalaryAccount : String
 - IsAuthrizer : Boolean
 - IsSecretary : Boolean
 - IsBooker : Boolean
 - IsFinance : Boolean
 - IsAdmin : Boolean
 - IsPreApprover : Boolean
 - IsFinanceTime : Boolean
 - IsEmployee : Boolean
 - IsError : Boolean
 - Errorstring : String
- ARRAYOFUSER
 - Count : Long
 - Position : Long
 - Array() : User
 - IsValid : Boolean

METODER OG KLASSER TIL HÅNDTERING AF EMS DATABASER

I modsætning til brugere, kan både dimensioner og bilag ligge i flere forskellige data siloer. Disse siloer kaldes EMS Databaser og benævnes i dette for en EMS.

Disse metoder og klasser benyttes til at finde den EMS der skal benyttes, når der arbejdes med dimensioner eller eksportdata. Der kan ikke arbejdes med disse to typer af data, uden at sende en EMS med i kaldet.

Metoder

- GETDEFAULTEMS()
- GETEMS(string)
- GETEMSUNITS()

Klasser

- EMS
 - EMSID : String
 - EMSName : String
 - EMSVersion : String
 - IsError : Boolean
 - Errorstring : String
- ARRAYOFEMS
 - Count : Long
 - Position : Long
 - Array() : User
 - IsValid : Boolean

METODER OG KLASSER TIL HÅNDTERING AF DIMENSIONER

Disse metoder og klasser benyttes til at hente, opdatere og oprette dimensioner.

Fælles for alle metoder er, at der skal sendes et EMS objekt med, for at fortælle metoden i hvilken EMS database man vil kikke i.

Informationer om brugeren bæres i en DIMENSION klasse, der returneres når man slår en dimension op, opretter en dimension eller opdatere en dimension.

Skal man modificere en dimension, slår man først dimensionen op og benytter derefter det DIMENSION objekt der returneres til at opdatere de data man ønsker. Til sidst returneres DIMENSION objektet og et opdateret DIMENSION objekt modtages retur, med eventuelle fejlkoder.

Skal man hente flere dimensioner, returneres en klasse ARRAYOF DIMENSION, der ud over et array med DIMENSION objekter også indeholder information om hvor mange der returneres og hvorfra i den samlede liste.

Metoder

- GETDIMENSION(Acubiz.EMS, string)
- GETDIMENSIONS(Acubiz.EMS, int, int)
- GETDIMENSIONSBYTYPE(Acubiz.EMS, int, int, string)
- CREATEDIMENSION(Acubiz.EMS, string, string, string, string, string, string, string)
- UPDATEDIMENSION(Acubiz.DIMENSION)

Klasser

- DIMENSION
 - ID : String
 - ParentID : String
 - DisplayValue : String
 - NameValue : String
 - ID2 : String
 - ID3 : String
 - IsError : Boolean
 - Errorstring : String
 - EMS : EMS
- ARRAYOFDIMENSION
 - Count : Long
 - Position : Long
 - Array() : User
 - IsValid : Boolean

METODER OG KLASSER TIL HÅNDTERING AF EKSPORT DATA

Metode

- EXPORTCSVDATA(Acubiz.EMS, string, string, bool)
- EXPORTXMLDATA(Acubiz.EMS, string, string, bool)

Klasse

Acubiz A/S

Bregnerødvej 133C

DK-3460 Birkerød

Tlf: +45 70 214 215

- EXPORTDATA
 - ID : String
 - Data : Variant
 - Count : Long
 - Format : String
 - ExportProfileName : String
 - ExportDateTime : String
 - IsExported : Boolean
 - IsError : Boolean
 - ErrorInfo : String

METODER OG KLASSER TIL HÅNDTERING AF IMPORT DATA

Metode

- IMPORTCSVDATA(Acubiz.EMS, string, string)
- IMPORTXMLDATA(Acubiz.EMS, string, string)

BRUGERE: LÆSNING, OPDATERING OG OPRETTELSE

Det er muligt at liste, opdatere og oprette brugere på systemet.

Det gøres via klassen USER og de funktioner der er tilknyttet SessionClient klassen (Se: Adgang via .NET, side 3).

LISTNING AF BRUGERE

Listning af brugere gøres gennem følgende funktioner:

- FINDUSER(string)
- GETUSERS(int, int)

FINDUSER

FINDUSER benyttes til at slå en given bruger op på initialer eller EmployeeID. Funktionen returnerer et USER objekt med data, hvis der blev fundet en bruger og et null objekt, hvis der ikke blev fundet en bruger.

1.1.1.2 Parametre

- Initialer (String): Initialer på den bruger man vil have returneret.

1.1.1.3 Retursvar

- USER objekt med data på den forespurgte bruger

1.1.1.4 Eksempel

Funktionen kaldes således:

```
Acubiz.USER User = WS.FINDUSER("initialer");
```

USER objektet der returneres kan modificeres og benyttes sammen med UpdateUser funktionen. Her gives det samme USER objekt som parameter til UpdateUser funktionen.

GETUSERS

GETUSERS benyttes til at finde alle brugere for en eventuel bulk håndtering.

Funktionen returnerer et array af USER objekter, der er pakket ind i objektet ARRAYOFUSERS.

1.1.1.5 Parametre

- Position (long): Hvor i den samlede brugerliste skal der udlæses fra. Angives 0 læses fra begyndelsen
- Count (long): Hvor mange elementer skal der udlæses

1.1.1.6 Retursvar

- ARRAYOFUSER objekt

1.1.1.7 Eksempel

Funktionen kaldes således:

```
Acubiz.ARRAYOFUSER Users = WS.GETUSERS(0, 0);
```

Efterfølgende kan antal af returnerede poster læses via `Users.COUNT`.

OPRETTELSE AF BRUGERE

Oprettelse af brugere gøres via følgende funktion:

- `CREATEUSER(string, string, string, string, string, bool, bool, bool)`

CREATEUSER

`CREATEUSER` opretter en bruger og returnerer det `USER` object der er oprettet, med de informationer der er angivet og de default værdier som det er konfigureret i systemet.

Det `USER` objekt der bliver returneret kan modificeres og bruges som parameter til `UPDATEUSER` funktionen, for at ændre/sætte de egenskaber som ikke bliver sat ved selve oprettelsen.

1.1.1.8 Parametre

- `Initials (String)` : Initialer på brugeren
- `Firstname (String)` : Fornavn på brugeren
- `Lastname (String)` : Efternavn på brugeren
- `Email (String)` : E-mail adresse til brugeren
- `EmployeeID (String)` : Brugerens medarbejdersnummer
- `IsAuthrizer (Boolean)` : Sættes til sand hvis brugeren er godkender
- `IsSecretary (Boolean)` : Sættes til sand hvis brugeren er sekretær
- `IsBooker (Boolean)` : Sættes til sand hvis brugeren er booker

1.1.1.9 Retursvar

- `USER` objekt med den oprettede brugere. Sker der fejl under oprettelsen, vil objektets `ISERROR` egenskab være sat til sand og `ERRORSTRING` vil indeholde den eller de fejl der blev fundet.

1.1.1.10 Eksempel

Funktionen bruges således:

Her gives input via `textBox` fra brugerfladen og sendes videre til oprettelsesfunktionen. Efterfølgende kontrolleres der for om der er sket en fejl.

Endeligt sættes `admin`, `finiance` og `preapprover` flagene på den nye bruger, som endelig opdateres. Retursvaret kontrolleres til sidst igen for fejl.

```
Acubiz.USER User = WS.CREATEUSER(    textBoxUserInitials.Text,
                                     textBoxUserGivenName.Text,
                                     textBoxUserSurName.Text,
                                     textBoxUserEmail.Text,
                                     "",
                                     checkBoxUserAuthorizer.Checked,
                                     checkBoxUserSecretary.Checked,
                                     checkBoxUserBooker.Checked);
```

```
if (User.ISERROR)
{
    MessageBox.Show("Error: " + User.ERRORSTRING, "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
User.ISADMIN = checkBoxUserAdmin.Checked;
User.ISFINANCE = checkBoxUserFinance.Checked;
User.ISPREAPPROVER = checkBoxUserPreApprover.Checked;

User = WS.UPDATEUSER(User);

if (User.ISERROR)
{
    MessageBox.Show(User.ERRORSTRING, "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

OPDATERING AF BRUGERE

Opdatering af brugere sker via funktionen:

- UPDATEUSER(Acubiz.USER)

UPDATEUSER

UPDATEUSER modtager et USER objekt med den bruger der skal opdateres i systemet.

USER objektet, der sendes til UPDATEUSER, skal tidligere være modtaget enten igennem brugeroprettelse eller via en opslagsfunktion. Har USER objektet ikke tidligere sin oprindelse fra API'ets andre funktioner, vil UPDATEUSER afvise opdateringen.

1.1.1.11 Parametre

- Bruger (USER): USER objektet fra den bruger der skal opdateres.

1.1.1.12 Retursvar

- Bruger (USER): Den opdaterede bruger med eventuelle fejl information.

1.1.1.13 Eksempel

Se Eksempel hos CREATEUSER side 11

DIMENSIONER: LÆSNING, OPDATERING OG OPRETTELSE

Det er muligt at læse, opdatere og oprette dimensioner i hver af de tilknyttede EMS database der er oprettet på systemet.

For at få adgang til dimensioner i EMS databaserne, skal man have et handle til den EMS database man vil interagere med.

OPSLAG EFTER EMS DATABASER

- GETDEFAULTEMS()
- GETEMS(string)
- GETEMSUNITS()

GETDEFAULTEMS

GETDEFAULTEMS returnerer den EMS database der er opsat til at være standard for den bruger der er logget in på SOAP API'et.

Det vil altså sige, at hvis man bruger SOAP API'et frontend

1.1.1.14 Parametre

- Funktionen modtager ingen parametre.

1.1.1.15 Returværdi

- EMS : Der returneres det EMS objekt der hører til den EMS database, som brugeren har som default EMS.

1.1.1.16 Eksempel

For at få fat i den authentifierede SOAP Api brugers EMS, kaldes følgende:

```
Acubiz.EMS ems = WS.GETDEFAULTEMS();
```

GETEMS

GETEMS returnerer den EMS database, der bliver forespurgt på som parameter til funktionen.

1.1.1.17 Parametre

- EmsID (String): ID identifikation på den EMS database der ønskes leveret retur.

1.1.1.18 Returværdi

- EMS : der returneres det EMS objekt der hører til EMS databasen, som der er angivet som parameter. Hvis der ikke blev fundet en database med det givne ID, returneres NULL.

1.1.1.19 Eksempel

For at få fat i en EMS database, med navnet "marketing", kaldes:

```
Acubiz.EMS ems = WS.GETEMS("marketing");
```

GETEMSUNITS

Funktionen returnerer alle registrerede EMS databaser I systemet.

1.1.1.20 Parametre

- Funktionen modtager ingen parametre.

1.1.1.21 Returværdi

- ARRAYOFEMS objekt

1.1.1.22 Eksempel

I dette eksempel hentes oplysninger om alle EMS databaser, hvorefter antallet skrives til en label og alle EMS databaser skrives til et listView.

Her lægges selve EMS objektet på listViewItem'ets tag parameter. Det vil sige at man kan tage direkte fat i EMS objektet, når listViewItem'et vælges, uden at skulle hente det igen.

```
listViewEMS.Items.Clear();
```

```
ListViewItem item = null;
```

```
Acubiz.ARRAYOFEMS EMSUnits = WS.GETEMSUNITS();  
labelEMSFound.Text = "EMS Found: " + EMSUnits.COUNT.ToString();  
foreach (Acubiz.EMS EMS in EMSUnits.ARRAY)  
{  
    item = new ListViewItem();  
    item.Text = EMS.EMSNAME;  
    item.Tag = EMS;  
    listViewEMS.Items.Add(item);  
}
```

LISTING AF DIMENSIONER

Til listing af dimensioner benyttes følgende funktioner:

- GETDIMENSION(Acubiz.EMS, string)
- GETDIMENSIONS(Acubiz.EMS, int, int)
- GETDIMENSIONSBYTYPE(Acubiz.EMS, int, int, string)

GETDIMENSION

GETDIMENSION finder og returnerer en enkelt dimension, på baggrund af dimensionens ID.

1.1.1.23 Parametre

- EMS (EMS): Handle til den EMS database der skal søges i
- ID (String): ID på den dimension der skal returneres

1.1.1.24 Returværdi

- DIMENSION objekt

1.1.1.25 Eksempel

I dette eksempel hentes der et EMS handle fra et listview (se eksempel fra GETEMSUNITS på side 14) og bruges til at finde den dimension, hvis ID er skrevet i textBox'en textBoxDimSearch.

```
Acubiz.EMS ems = (Acubiz.EMS)listViewEMS.SelectedItems[0].Tag;  
Acubiz.DIMENSION dimension = WS.GETDIMENSION (ems, textBoxDimSearch.Text);
```

I dette eksempel hentes dimensionen "0010" I EMS databasen "marketing".

```
Acubiz.EMS ems = WS.GETEMS("marketing");  
Acubiz.DIMENSION dimension = WS.GETDIMENSION (ems, "0010");
```

GETDIMENSIONS

GETDIMENSIONS henter alle dimensioner, der ligger i den EMS database der er angivet som parameter.

1.1.1.26 Parametre

- EMS (EMS): Handle til den EMS database der skal søges i
- Position (long): Hvor i den samlede liste af dimensioner skal der udlæses fra. Angives 0 læses fra begyndelsen
- Count (long): Hvor mange elementer skal der udlæses

1.1.1.27 Returværdi

- ARRAYOFDIMENSIONS objekt

1.1.1.28 Eksempel

Dette eksempel bygger videre på vores tidligere listView as EMS databaser og populerer et listview med de dimensioner der bliver returneret.

Den valgte EMS database i listViewEMS findes og bruges som ems database til at få alle dimensioner.

Efterfølgende skrives antallet af dimensioner i en label og hver dimension gennemgås og skrives til et listView.

```
Acubiz.EMS ems = (Acubiz.EMS)listViewEMS.SelectedItems[0].Tag;  
Acubiz.ARRAYOFDIMENSION dimensions = WS.GETDIMENSIONS(ems, 0, 0);  
labelDimFound.Text = "Dimensions found: " + dimensions.COUNT.ToString();
```

```
ListViewItem item = null;
```

```
listViewDim.Items.Clear();
```

```
string[] arr = new string[5];
```

```
foreach (Acubiz.DIMENSION dimension in dimensions.ARRAY)  
{  
    arr[0] = dimension.ID;  
    arr[1] = dimension.NAMEVALUE;
```

```
arr[2] = dimension.DISPLAYVALUE;
arr[3] = dimension.ID2;
arr[4] = dimension.ID3;

item = new ListViewItem(arr);
listViewDim.Items.Add(item);
}
```

```
}
```

Dette eksempel henter alle dimensioner i EMS databasen "marketing":

```
Acubiz.EMS ems = WS.GETEMS("marketing");
```

```
Acubiz.ARRAYOFDIMENSION dimensions = WS.GETDIMENSIONS(ems, 0, 0);
```

GETDIMENSIONSBYTYPE

GETDIMENSIONS henter alle dimensioner af en bestemt type, der ligger i den EMS database der er angivet som parameter.

1.1.1.29 Parametre

- EMS (EMS): Handle til den EMS database der skal søges i
- Position (long): Hvor i den samlede liste af dimensioner skal der udlæses fra. Angives 0 læses fra begyndelsen
- Count (long): Hvor mange elementer skal der udlæses
- Type (string): Den dimensionstype der skal hentes ind. Dette er typisk ID på det øverste niveaus dimensioner.

1.1.1.30 Returværdi

- ARRAYOFDIMENSIONS objekt

1.1.1.31 Eksempel

Dette eksempel henter alle dimensioner i EMS databasen "marketing", der har typen "Dim1":

```
Acubiz.EMS ems = WS.GETEMS("marketing");
```

```
Acubiz.ARRAYOFDIMENSION dimensions = WS.GETDIMENSIONS(ems, 0, 0, "Dim1");
```

OPRETTELSE AF DIMENSIONER

Til oprettelse af en dimension benyttes følgende funktion:

- CREATEDIMENSION(Acubiz.EMS, string, string, string, string, string, string)

CREATEDIMENSION

CREATEDIMENSION opretter en bruger og returnerer det DIMENSION objekt der er oprettet, med de informationer der er angivet.

Det DIMENSION objekt der bliver returneret kan modificeres og bruges som parameter til UPDATE DIMENSION funktionen, for at ændre/sætte de egenskaber som ikke bliver sat ved selve oprettelsen.

1.1.1.32 Parametre

- EMS (EMS) : Handle til den EMS database der skal oprettes i
- ID (String) : ID på den dimension der skal oprettes. ID skal være unik i EMS databasen
- Parent ID (String) : ID på den dimension der ligger oven over
- Name (String) : Navn på dimension
- Display (String) : Navn til visning af dimension
- ID2 (String) : 2. nøgle på dimensionen
- ID3 (String) : 3. nøgle på dimensionen

1.1.1.33 Retursvar

- DIMENSION objekt med den oprettede brugere. Sker der fejl under oprettelsen, vil objektets ISERROR egenskab være sat til sand og ERRORSTRING vil indeholde den eller de fejl der blev fundet.

1.1.1.34 Eksempel

Funktionen bruges således:

```
Acubiz.DIMENSION dimension = WS.CREATEDIMENSION(ems, textBoxDimID.Text, textBoxDimLevel.Text, textBoxDIMName.Text, textBoxDimDisplay.Text, textBoxDimID2.Text, textBoxDIMID3.Text);
```

OPDATERING AF DIMENSIONER

Opdatering af dimensioner sker via funktionen:

- UPDATEDIMENSION(Acubiz.DIMENSION)

UPDATEDIMENSION

UPDATEDIMENSION modtager et DIMENSION objekt med den dimension der skal opdateres i systemet.

DIMENSION objektet, der sendes til UPDATEDIMENSION, skal tidligere være modtaget enten igennem oprettelse eller via en opslagsfunktion. Har DIMENSION objektet ikke tidligere sin oprindelse fra API'ets andre funktioner, vil UPDATEDIMENSION afvise opdateringen.

1.1.1.35 Parametre

- Dimension (DIMENSION): DIMENSION objektet fra den dimension der skal opdateres.

1.1.1.36 Retursvar

- Dimension (DIMENSION): Den opdaterede dimension med eventuelle fejl information.

1.1.1.37 Eksempel

```
Acubiz.DIMENSION dimension = WS.GETDIMENSION(ems, ID);  
dimension = WS.UPDATEDIMENSION(dimension);
```

IMPORT AF DATA

Import af data foregår ud fra import profiler der allerede er oprettet i systemet.

Import af data bruges til at sende mange data op på samme tid. Her fås dog ikke fejlmeddelelser eller valideringsfejl tilbage, ud over den validering der foregår på om det format der sendes op er korrekt.

IMPORTCSVDATA

IMPORTCSVDATA sætter en dataimport i gang og returnerer derefter en indikering om datasættet blev accepteret til import eller ej.

Et data sæt afvises, hvis Import Profilen er forkert, hvis formatet er forkert eller hvis der mangler felter eller poster i dataformatet.

1.1.1.38 Parametre

- ImportProfile (String) : Eksport profil navn
- ImportType (String) : Eksport type
- ImportData (String) : Import data I CSV format

Import formatet er i CSV format, og fremgår af den pågældende opsætningsprofil

1.1.1.39 Retursvar

- Boolean: Hvis ImportData blev accepteret til import og importen er sat I gang. Dette er ikke en indikering om selve importen er succesfuld eller ej.

1.1.1.40 Eksempel

```
Acubiz.EMS ems = WS.GETEMS("marketing");  
Acubiz.EXPORTDATA IMPORTCSVDATA = WS.IMPORTCSVDATA(ems, textBoxImportProfile.Text, textBoxImportType.Text, CSV);
```

IMPORTXMLDATA

IMPORTXMLDATA sætter en dataimport i gang og returnerer derefter en indikering om datasættet blev accepteret til import eller ej.

Et data sæt afvises, hvis Import Profilen er forkert, hvis formatet er forkert eller hvis der mangler felter eller poster i dataformatet.

1.1.1.41 Parametre

- ImportProfile (String) : Eksport profil navn
- ImportType (String) : Eksport type
- ImportData (String) : Import data I XML format

Import formatet er i XML format, og fremgår af den pågældende opsætningsprofil

1.1.1.42 Retursvar

- Boolean: Hvis ImportData blev accepteret til import og importen er sat i gang. Dette er ikke en indikering om selve importen er succesfuld eller ej.

1.1.1.43 Eksempel

```
Acubiz.EMS ems = WS.GETEMS("marketing");  
Acubiz.EXPORTDATA IMPORTXMLData = WS.IMPORTCSVDATA(ems, textBoxImportProfile.Text, textBoxImportType.Text, XML);
```

EKSPORT AF DATA

Eksport af data foregår ud fra eksport profiler der allerede er oprettet i systemet.

Ligesom man kan få eksporterne ud som filer, kan man også få dem ud via SOAP API'et. Det gøres via funktionen EKSPORTCSVDATA og EKSPORTXMLDATA.

EXPORTCSVDATA

EKSPORTCSVDATA sætter en dataeksport i gang og leverer efterfølgende et dataset af data retur.

1.1.1.44 Parametre

- EMS (EMS) : Handle til den EMS database der skal oprettes i
- ExportProfile (String) : Eksport profil navn
- ExportType (String) : Eksport type
- IsTestExport (Bool) : Angiver om det er en test eksport eller om data skal markeres som eksporteret.

1.1.1.45 Retursvar

- EXPORTCSVDATA objekt

Data returneres i et EXPORTDATA objekt, der holder data samt informationer om eventuelle fejl, antal og et unikt ID på eksporten:

- EXPORTCSVDATA
 - ID : String
 - Data : Variant
 - Count : Long
 - Format : String
 - ExportProfileName : String
 - ExportDateTime : String
 - IsExported : Boolean
 - IsError : Boolean
 - ErrorInfo : String

Eksporten returneres med et ID, der er en unik identifikator på denne eksport.

Data ligger i DATA varianten, som et streng array af datalinjer. Datalinjerne er i denne version af API'et er semikolon separerede linjer med de felter som der er beskrevet i eksport profilen. Man kan altså selv bestemme hvilke felter der skal returneres i eksport profilen, der sættes op i applikationen.

Fejl returneres i ErrorInfo, hvor IsError er sat til sand, hvis det er en fejl.

Hvis eksporten er fuldendt, sættes IsExported til sand. Er der opstået et problem, der gør at datasættet ikke er eksporteret, sættes IsExported til falsk.

Format returneres som CSV i dette kald.

Count indeholder antal linjer i eksporten.

1.1.1.46 Eksempel

Dette eksempel henter alle de eksportdata, der ikke er eksporteret i EMS databasen "marketing", fra eksport profilen "01. Omkostninger - uden summering" med eksport typen "Expense Accounts".

```
Acubiz.EMS ems = WS.GETEMS("marketing");
```

```
Acubiz.EXPORTDATA exportData = WS.EXPORTCSVDATA(ems,"01. Omkostninger - uden summering", "Expense Accounts", true);
```

EXPORTXMLDATA

EXPORTXMLDATA sætter en dataeksport i gang og leverer efterfølgende et dataset af data retur.

1.1.1.47 Parametre

- EMS (EMS) : Handle til den EMS database der skal oprettes i
- ExportProfile (String) : Eksport profil navn
- ExportType (String) : Eksport type
- IsTestExport (Bool) : Angiver om det er en test eksport eller om data skal markeres som eksporteret.

1.1.1.48 Retursvar

- EXPORTXMLDATA objekt

Data returneres i et EXPORTDATA objekt, der holder data samt informationer om eventuelle fejl, antal og et unikt ID på eksporten:

- EXPORTXMLDATA
 - ID : String
 - Data : Variant
 - Count : Long
 - Format : String
 - ExportProfileName : String
 - ExportDateTime : String
 - IsExported : Boolean
 - IsError : Boolean
 - ErrorInfo : String

Eksporten returneres med et ID, der er en unik identifikator på denne eksport.

Acubiz A/S

Bregnerødvej 133C

DK-3460 Birkerød

Tlf: +45 70 214 215

Data ligger i DATA varianten, som et streng array af XML data. Datalinjerne er i denne version af API'et er XML med de felter som der er beskrevet i eksport profilen. Man kan altså selv bestemme hvilke felter der skal returneres i eksport profilen, der sættes op i applikationen.

Fejl returneres i ErrorInfo, hvor IsError er sat til sand, hvis det er en fejl.

Hvis eksporten er fuldendt, sættes IsExported til sand. Er der opstået et problem, der gør at datasættet ikke er eksporteret, sættes IsExported til falsk.

Format sættes til XML ved dette kald.

Count indeholder antal linjer i eksporten.

1.1.1.49 Eksempel

Dette eksempel henter alle de eksportdata, der ikke er eksporteret i EMS databasen "marketing", fra eksport profilen " 01. Omkostninger - uden summering" med eksport typen " Expense Accounts".

```
Acubiz.EMS ems = WS.GETEMS("marketing");
```

```
Acubiz.EXPORTDATA exportData = WS.EXPORTXMLDATA(ems," 01. Omkostninger - uden summering", " Expense Accounts", true);
```

INDEX

A

APIVERSION; 6
ARRAYOFDIMENSION; 8
ARRAYOFEMS; 7; 14
ARRAYOFUSER; 7

B

Brugere: Læsning, opdatering og oprettelse; 10

C

CREATEDIMENSION; 8; 16
CREATEUSER; 6; 11; 16

D

DIMENSION; 8

E

EMS; 7
ERRORSTRING; 6
EXPORTDATA; 9; 18; 19; 20; 21

F

FINDUSER; 6; 10

G

GETCURRENTUSER; 6

GETDEFAULTEMS; 7; 13

GETDIMENSION; 8; 14

GETDIMENSIONS; 8; 14; 15

GETDIMENSIONSBYTYPE; 8; 14; 16

GETEFFECTIVEUSERNAME; 6

GETEMS; 7; 13

GETEMSUNITS; 7; 13

GETEXPORTDATA; 8; 9

GETUSERNAME; 6

GETUSERS; 6; 10

I

ISERROR; 6

L

Listning af brugere; 10

O

Opdatering af brugere; 12; 17

Oprettelse af brugere; 11

U

UPDATEDIMENSION; 8

UPDATEUSER; 6; 12; 17

USER; 6