

Versjon: 1.0
Dato: 16.10.14



evry.com

Cerebrum2Ephorte – en Ephorte web-tjeneste for Cerebrum

Implementert hos Universitet i Oslo.

Robin Berg Pettersen / Frank Sandersen

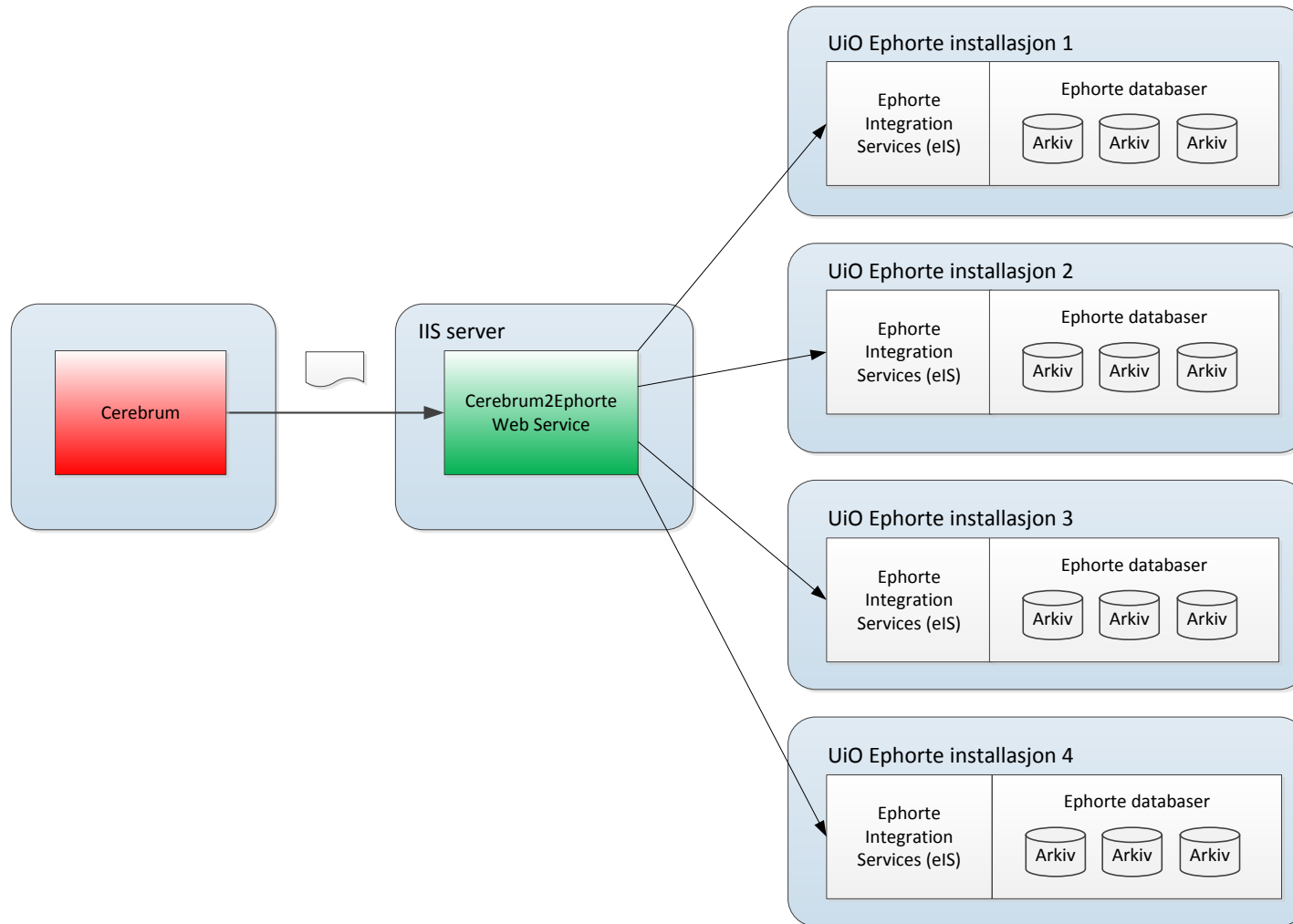
Systemutviklere

Product Development & Maintenance

robin.berg.pettersen@evry.com / frank.sandersen@evry.com

1 Introduksjon

Cerebrum2Ephorte er en Ephorte web service skreddersydd for Cerebrum. Cerebrum benyttes for sentralisert tilgangsadministrasjon hos UiO. Cerebrum2Ephorte har web tjenester som er nødvendig for at Cerebrum kan utføre relevant brukeradministrasjon i flere Ephorte-miljøer slik som vist i skissen nedenfor.



Hver metode i web servicen har kunde som input parameter for å skille på de forskjellige Ephorte installasjonene hos UiO. I tillegg må man alltid oppgi navn på ephorte database innenfor den enkelt Ephorte installasjon. Det vil dermed være Cerebrum som holder orden på hvilken kunde, database og bruker som skal behandles. Web Servicen gir alltid tilbakemelding på om operasjoner går bra og sender med eventuelle feilmeldinger slik at Cerebrum kan håndtere disse med eventuelt mailvarsling til UiO om dette er ønskelig. Web Servicen er laget mest mulig generisk slik at regler (forretningslogikk) vedlikeholdes av Cerebrum samtidig som det er minimalt konfigurasjon av web servicen. Nødvendig konfigurasjon vil alltid være en del av input parameterne til web metodene.

1.1 Bruk av Ephorte Integration Services (EIS)

Cerebrum2Ephorte fungerer mot én eller flere ephorte installasjoner/servere samt én eller flere databaser innenfor den enkelte installasjon. Nye installasjoner og databaser må legges inn i konfigurasjonen med tilhørende ephorte bruker som skal benyttes mot disse.

Cerebrum2Ephorte benytter EIS (Ephorte Integration Services) for kommunikasjon med Ephorte. Følgende EIS-tjenester benyttes:

Service	Relative service url	WCF endpoint navn	Kommentar
DocumentService v3	.../services/documents/v3/DocumentService.svc	DocumentServiceV3_BasicHttpBinding	
ObjectModelService v3 No	.../services/objectmodel/v3/No/ObjectModelService.svc	ObjectModelServiceV3No_WSHttpBinding	
ObjectModelService v1	.../services/objectmodel/v1/ObjectModelService.svc/ws	ObjectModelServiceV1_WSHttpBinding	Benyttet unntaksvis på grunn av Ephorte4 v2.0.19 på enkelte baser.
FunctionService v1	.../services/Functions/v1/FunctionsService.svc/ws	FunctionsServiceV1_WSHttpBinding	

ObjectModelService v3 har ikke vært skikkelig støttet i Ephorte før Ephorte4 v2.0.23 og senere. Om man skulle ha gamle baser på Ephorte4 v2.0.19 kan enkelte funksjoner implementeres med ObjectModelService v1.

1.2 Kildekode

Kildekode til Cerebrum2Ephorte finnes hos EVERY med følgende referanse:

TFS Server: <http://tfs.egdev.net:8080/tfs/>

TFS Collection: DefaultCollection

TFS Project: UiO.CerebrumWS

2 Web metoder

Dette kapitlet inneholder de forskjellige web metoder i Cerebrum2Ephorte – metodenavn, beskrivelse og .net kode-eksempel som viser bruk.

```
[ServiceContract(Name = "Cerebrum2EphorteService", Namespace = "http://Cerebrum2Ephorte/Service")]
public interface IService
{
    [OperationContract]
    DTO.TestUser Test(string customer, string userId);
    [OperationContract]
    DTO.TestUser TestWithEphorte(string customerId, string database, string userId);
    [OperationContract]
    DTO.EphorteOrgUnits GetAllOrgUnits(string customerId, string database);
    [OperationContract]
    DTO.EphorteRoles GetAllRoles(string customerId, string database);
    [OperationContract]
    DTO.EphorteAccessCodes GetAllAccessCodes(string customerId, string database);
    [OperationContract]
    DTO.EphorteUsers GetAllUsers(string customerId, string database);
    [OperationContract]
    DTO.EphorteUserDetails GetUserDetails(string customerId, string database, string userId);
    [OperationContract]
    DTO.Response EnsureUser(string customerId, string database, DTO.EphorteUser user);
    [OperationContract]
    DTO.Response DisableUser(string customerId, string database, string userId);
    [OperationContract]
    DTO.Response EnsureRoleForUser(string customerId, string database, string userId, string jobTitle, string roleId, string orgId,
                                   string fondsSeriesId, string registryManagementUnitId, bool? setAsDefaultRole);
    [OperationContract]
    DTO.Response EnsureAccessCodeAuthorizationForUser(string customerId, string database, string userId, string accessCodeId,
                                                       string orgId, bool isAuthorizedForAllUnits);
    [OperationContract]
    DTO.EphorteUserBacklog GetUserBacklog(string customerId, string database, string userId);
    [OperationContract]
    DTO.Response DisableRolesAndAuthorizationsForUser(string customerId, string database, string userId);
}
```

2.1 Metode: Test

Hensikten med metoden er å teste at web servicen fungerer uten at man oppdaterer noe i Ephorte. Metoden forventer userId=X00059. Hvis userId angis til noe annet vil det returneres en feilmelding.

Eksempel:

```
var ephorteUserTest = proxy.Test(«UiO», "X00059");
if (!ephorteUserTest.HasError)
{
    Console.WriteLine(string.Format("User {0} found", ephorteUserTest.UserId));
}
else
    Console.WriteLine(string.Format("{0}", ephorteUserTest.ErrorMessage));
```

2.2 Metode: TestWithEphorte

Hensikten med denne metoden er å teste web servicen og samtidig gjøre et reelt oppslag mot brukertabellen i Ephorte. Her må man angi en gyldig Ephorte database og et gyldig Ephorte brukernavn.

Eksempel:

```
var ephorteUserTest = proxy.TestWithEphorte(«UiO», «EPHORTEK», "BJOJO");
if (!ephorteUserTest.HasError)
{
    Console.WriteLine(string.Format("User {0} found", ephorteUserTest.UserId));
}
else
    Console.WriteLine(string.Format("{0}", ephorteUserTest.ErrorMessage));
```

2.3 Metode: GetAllOrgUnits

Denne metoden returnerer alle aktive (ikke nedlagte) organisasjonsenheter fra Ephorte. Brukere er ikke direkte koblet til organisasjonsenheter men deres rolletilganger og autorisasjoner kan være det.

Eksempel:

```
var ephorteOrgUnits = proxy.GetAllOrgUnits(«UiO», «EPHORTEK»);
if (!ephorteOrgUnits.HasError)
{
    foreach (EphorteOrg org in ephorteOrgUnits.OrgUnits)
        Console.WriteLine(string.Format("Org found: {0} ({1}) - {2}", org.OrgId, org.Name, org.IsTop ? "Top org" : "Not top org"));
}
else
    Console.WriteLine(string.Format("{0}", ephorteOrgUnits.ErrorMessage));
```

2.4 Metode: GetAllRoles

Denne metoden returnerer alle roller. Disse benyttes sammen med organisasjonsenheter når man skal tilordne rolletilganger til en bruker.

Eksempel:

```
var ephorteRoles = proxy.GetAllRoles(«UiO», «EPHORTEK»);
if (!ephorteRoles.HasError)
{
    foreach (EphorteRole role in ephorteRoles.Roles)
        Console.WriteLine(string.Format("Role found: {0} - {1}", role.RoleId, role.Description));
}
else
    Console.WriteLine(string.Format("{0}", ephorteRoles.ErrorMessage));
```

2.5 Metode: GetAllAccessCodes

Denne metoden returnerer alle aktive tilgangskoder. Disse benyttes sammen med organisasjonsenheter når man skal gi en bruker en autorisasjon.

Eksempel:

```
var ephorteAccessCodes = proxy.GetAllAccessCodes(«UiO», «EPHORTEK»);
if (!ephorteAccessCodes.HasError)
{
    foreach (EphorteAccessCode accesscode in ephorteAccessCodes.AccessCodes)
        Console.WriteLine(string.Format("Access code found: {0} - {1}", accesscode.AccessCodeId, accesscode.Description));
}
else
    Console.WriteLine(string.Format("{0}", ephorteAccessCodes.ErrorMessage));
```

2.6 Metode: GetAllUsers

Denne metoden returnerer en liste med alle aktive (gyldige) brukere. Hver bruker inneholder kontaktinformasjon. Skal man ha mer informasjon om den enkelte bruker må man videre kalle GetUserDetails() for den enkelte bruker.

Eksempel:

```
var ephorteUsers = proxy.GetAllUsers(«UiO», «EPHORTEK»);
if (!ephorteUsers.HasError)
{
    Console.WriteLine(string.Format("Users found: {0}", ephorteUsers.Users.Count()));
    foreach (EphorteUser user in ephorteUsers.Users)
    {
        Console.WriteLine(string.Format("User found: {0} - {1}", user.UserId, user.FullName));

        var ephorteUser = proxy.GetUserDetails(«UiO», «EPHORTEK», user.UserId);
        if (!ephorteUser.HasError)
        {
            Console.WriteLine(string.Format("Details for user {0} found: {1} roles and {2} authorizations", ephorteUser.User.UserId,
ephorteUser.UserRoles.Count(), ephorteUser.UserAuthorizations.Count()));
        }
        else
            Console.WriteLine(string.Format("{0}", ephorteUser.ErrorMessage));
    }
}
else
    Console.WriteLine(string.Format("{0}", ephorteUsers.ErrorMessage));
```

2.7 Metode: GetUserDetails

Denne metoden returnerer kontaktinformasjon om bruker samt lister med brukers rolletilganger og autorisasjoner.

Eksempel:

```
var ephorteUser = proxy.GetUserDetails(«UiO», «EPHORTEK», "BJOJO");
if (!ephorteUser.HasError)
{
    Console.WriteLine(string.Format("Details for user {0} found: {1} roles and {2} authorizations", ephorteUser.User.UserId,
    ephorteUser.UserRoles.Count(), ephorteUser.UserAuthorizations.Count()));
}
else
    Console.WriteLine(string.Format("{0}", ephorteUser.ErrorMessage));
```

2.8 Metode: EnsureUser

Denne metoden oppretter bruker hvis denne ikke finnes fra før. Hvis bruker finnes fra før vil kontaktinformasjon oppdateres.

Eksempel:

```
var newUser = new EphorteUser
{
    UserId = "OLANOR5",
    FirstName = "Ola",
    LastName = "Nordmann",
    FullName = "Ola Nordmann",
    EmailAddress = "ola.nordmann@norge.no",
    Telephone = "12345678",
    Mobile = "99911999",
    StreetAddress = "Postveien 1",
    ZipCode = "3960",
    City = "Stathelle",
};
var response = proxy.EnsureUser(«UiO», «EPHORTEK», newUser);
if (!response.HasError)
    Console.WriteLine(string.Format("User created/updated: {0}", newUser.UserId));
else
    Console.WriteLine(string.Format("{0}", response.ErrorMessage));
```

2.9 Metode: EnsureRoleForUser

Denne metoden oppretter/oppdaterer en personlig rolle for en bruker. Personlige roller kobles blant annet mot til organisasjonsenhet, journalenhet og rolle.

Eksempel:

```
var response = proxy.EnsureRoleForUser(«UiO», «EPHORTEK», "KARNOR", "Arkivansvarlig", "AR1", "ADM", "SAK1", "JOURNAL", false);
if (!response.HasError)
    Console.WriteLine(string.Format("User role created/updated: {0}", "Arkivansvarlig"));
else
    Console.WriteLine(string.Format("{0}", response.ErrorMessage));
```

2.10 Metode: EnsureAccessCodeAuthorizationForUser

Denne metoden oppretter/oppdaterer en autorisasjon for en bruker. Autorisasjon er en klarering for en bestemt tilgangskode i Ephorte. Autorisasjonen kan enten gjelde for hele virksomheten, kun egne saker eller saker under en bestemt orgenhet.

- For kun egne saker settes orgId=null og isAuthorizedForAllUnits=false
- For hele virksomheten settes orgId=null og isAuthorizedForAllUnits=true
- For saker under bestemt orgenhet settes orgId=<kortnavn for administrativ enhet> og isAuthorizedForAllUnits=false

Eksempler:

```
//Ola Nordmann: UO for saker i hele virksomheten (kommunen) - NIVÅ 2
var response1 = proxy.EnsureAccessCodeAuthorizationForUser(«UiO», "EPHORTE-FK", "OLANOR", "UO", null, true);
if (response1.HasError) Console.WriteLine(string.Format("{0}", response1.ErrorMessage));

//Ola Nordmann: E for saker i egen enhet (FSK) - NIVÅ 1
response1 = proxy.EnsureAccessCodeAuthorizationForUser(«UiO», «EPHORTEK», "OLANOR", "E", "FSK", false);
if (response1.HasError) Console.WriteLine(string.Format("{0}", response1.ErrorMessage));

//Ola Nordmann: P for egne saker - NIVÅ 0
response1 = proxy.EnsureAccessCodeAuthorizationForUser(«UiO», «EPHORTEK», "OLANOR", "P", null, false);
if (response1.HasError) Console.WriteLine(string.Format("{0}", response1.ErrorMessage));
```

2.11 Metode: GetUserBacklog

Denne metoden benyttes for å hente informasjon om åpne saker for en bruker. Informasjonen kan f.eks presenteres i en mail til vedkommende med kopi til han/hennes leder en tid før man skal bytte jobb eller slutte. Metoden henter følgende informasjon om bruker:

- alle åpne saker der bruker er saksansvarlig og saksstatus er R, B eller V
- alle åpne journalposter der bruker er saksbehandler og journalstatus er R eller M

Ingen meldinger betyr ingen backlogg (HasBacklog = false). Metoden returnerer også email adresse til bruker og leder (= den som har LD rolle mot admin-enhet tilknyttet brukers standard rolle) som kan benyttes til utsendelse av mail.

Eksempel:

```
var userBackLog = proxy.GetUserBacklog(«UiO», «EPHORTEK», "BJOJO");
if (!userBackLog.HasError)
{
    if (userBackLog.HasBacklog)
    {
        foreach (string logMsg in userBackLog.BacklogMessage)
            Console.WriteLine(logMsg);

        Console.WriteLine(string.Format("User email: {0}", userBackLog.UserEmail));
        Console.WriteLine(string.Format("Leader email: {0}", userBackLog.LeaderEmail));
    }
}
else
    Console.WriteLine(string.Format("{0}", userBackLog.ErrorMessage));
```

Output fra denne metoden kan være slik:

```
"BJOJO er saksansvarlig for sak 2010/1719 som har status B"
"BJOJO er saksansvarlig for sak 2010/1620 som har status B"
"BJOJO er saksansvarlig for sak 2010/1597 som har status B"
"BJOJO er saksansvarlig for sak 2010/133 som har status B"
"BJOJO er saksbehandler for journalpost 127/2012 i sak 2012/25. Journalstataus er R"
"BJOJO er saksbehandler for journalpost 122/2012 i sak 2010/133. Journalstataus er R"
"BJOJO er saksbehandler for journalpost 120/2012 i sak 2011/241. Journalstataus er R"
"BJOJO er saksbehandler for journalpost 119/2012 i sak 2011/19. Journalstataus er R"
"BJOJO er saksbehandler for journalpost 118/2012 i sak 2011/19. Journalstataus er R"
"BJOJO er saksbehandler for journalpost 589/2011 i sak 2010/1965. Journalstataus er R"
"BJOJO er saksbehandler for journalpost 235/2011 i sak 2010/2013. Journalstataus er R"
"BJOJO er saksbehandler for journalpost 4651/2010 i sak 2010/1719. Journalstataus er R"
"BJOJO er saksbehandler for journalpost 958/2011 i sak 2011/314. Dokumenttype er I og journalposten er ikke avskrevet!"
"BJOJO er saksbehandler for journalpost 677/2011 i sak 2011/241. Dokumenttype er I og journalposten er ikke avskrevet!"
"BJOJO er saksbehandler for journalpost 609/2011 i sak 2011/19. Dokumenttype er I og journalposten er ikke avskrevet!"
```

2.12 Metode: DisableUser

Denne metoden vil deaktivere angitt bruker slik at denne ikke kan benyttes for innlogging.

Eksempel:

```
var response = proxy.DisableUser(«UiO», «EPHORTEK», "KARNOR");
if (!response.HasError)
    Console.WriteLine(string.Format("User deactivated: {0}", "KARNOR"));
else
    Console.WriteLine(string.Format("{0}", response.ErrorMessage));
```

2.13 Metode: DisableRolesAndAuthorizationsForUser

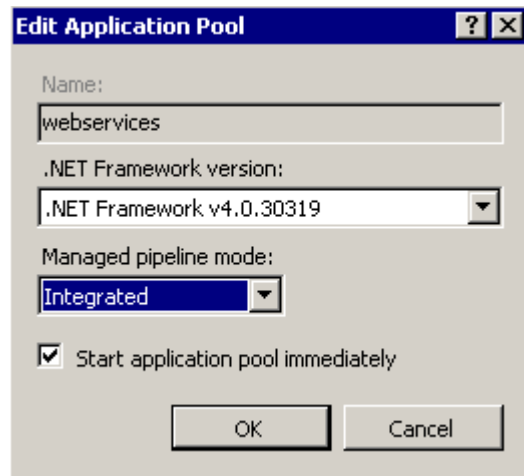
Denne metoden vil deaktivere alle personroller og autorisasjoner på angitt bruker.

Eksempel:

```
var response = proxy.DisableRolesAndAuthorizationsForUser(«UiO», «EPHORTEK», "OLANOR");
if (response.HasError) Console.WriteLine(string.Format("{0}", response.ErrorMessage));
```

3 Installasjon

Her følger en kort beskrivelse av installasjon av web servicen Cerebrum2Ephorte. Den betinger at man har web servicen tilgjengelig som pakket web applikasjon (zip-fil). Den nye servicen skal bruke .Net 4.0 app pool:



Authentication: Anonymous Authentication Enabled og resten av alternativene kan være Disabled.

Installasjonsprosedyre

1. Opprett ny katalog for web site
Eksempel på plassering: c:\inetpub\wwwroot\Cerebrum2EphorteService
2. Kopier zip-filen Cerebrum2EphorteWebApp.zip til server. Start InetMgr og naviger ned til Sites -> Default web . Høyreklikk på denne og velg «Deploy->Import Application»
3. Test ny web site ved å legge inn gjeldende url i browser og se at det åpnes en web service beskrivelse i browseren.
F.eks denne url: <http://ephuiotest.trofast.uio.no/Cerebrum2Ephorte/Service.svc>

4 Konfigurasjon

Konfigurasjon av servicen gjøres i filen web.config.

Konfig av Ephorte installasjon per kunde

```
<customConfiguration>
  <commonConfig smtpserver="159.171.252.75"
    tempfolder="%TEMP%\Cerebrum2EphorteService\_tempFiles"
    databasesWithNoPersonAddressSupport="EPHORTE-OLD1,EPHORTE-OLD2" />
  <ephorteConfigurations>
    <clear/>
    <!--serviceBaseUrl ends with /ePhorteWeb/ or /NCore/-->
    <add id="Ui0" desc="Ui0 Ephorte" user="CerebrumAdm" password="..." role="" serviceBaseUrl="http://ephuiotest1/ePhorteWeb/" />
    <add id="Ui02" desc="Ui0 Ephorte 2" user="CerebrumAdm" password="....." role="" serviceBaseUrl="http://ephuiotest2/ePhorteWeb/" />
    <add id="Ui03" desc="Ui0 Ephorte 3" user="CerebrumAdm" password="....." role="" serviceBaseUrl="http://ephuiotest3/ePhorteWeb/" />
  </ephorteConfigurations>
</customConfiguration>
```

Forklaring til `commonConfig->databasesWithNoPersonAddressSupport`: Unntasvis kan det finnes gamle baser som ikke er riktig oppgradert. Disse støtter ikke registrering av personadresser via EIS

Forklaring til `commonConfig->smtpserver`: Denne brukes foreløpig ikke.

Forklaring til `commonConfig->tempfolder`: Denne brukes foreløpig ikke men den ikke være tom.

Forklaring til `ephorteConfigurations->add`: Her legger man inn ny kunde. `id` er kundeid som også brukes ved kall mot web metodene. `user` / `password` / `role` angir den Ephorte brukeren som benyttes av web servicen ved arbeid i Ephorte. `serviceBaseUrl` er root url som sammen med EIS service url vil danne en komplett url.

Logging med standard komponent Log4Net

```
<log4net>
  <appender name="RollingFile" type="log4net.Appender.RollingFileAppender">
    <!-- App Pool exec account need write access to web site folder to create log files-->
    <!--<file value="{TEMP}\Cerebrum2EphorteService\Service.log"/>-->
    <file value="c:\windows\temp\Cerebrum2EphorteService\Service.log"/>
    <appendToFile value="true"/>
    <maximumFileSize value="10000KB"/>
    <maxSizeRollBackups value="25"/>
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date %C{1}.%M - %message%newline"/>
      <!--<conversionPattern value="%-5p [%t] %C{1}.%M - %m%n"/>-->
    </layout>
  </appender>
  <appender name="SmtpAppender" type="log4net.Appender.SmtpAppender">
    <to value="robin.berg.pettersen@evry.com"/>
    <from value="Cerebrum2EphorteService@usit.uio.no"/>
    <subject value="Cerebrum2EphorteService Error"/>
    <smtpHost value="159.171.252.75"/>
    <bufferSize value="1"/>
    <lossy value="true"/>
    <evaluator type="log4net.Core.LevelEvaluator">
      <threshold value="ERROR"/>
    </evaluator>
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%newline%date [%thread] %-5level %logger [%property{NDC}] - %message%newline%newline%newline"/>
    </layout>
  </appender>
  <appender name="ConsoleAppender" type="log4net.Appender.ConsoleAppender">
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date [%thread] %-5level %logger [%ndc] - %message%newline"/>
    </layout>
  </appender>
  <appender name="EventLogAppender" type="log4net.Appender.EventLogAppender">
    <applicationName value="Cerebrum2Ephorte Service"/>
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%newline"/>
    </layout>
  </appender>
</root>
  <!-- use WARN in production -->
  <!--<level value="WARN"/>-->
  <level value="DEBUG"/>
  <appender-ref ref="RollingFile"/>
```

```
<appender-ref ref="EventLogAppender"/>  
<appender-ref ref="SmtpAppender"/>  
</root>  
</log4net>
```

5 Driftsinformasjon

Loggfilene kan etter hvert bli mange så lenge logg-level = DEBUG. Det er mulig å justere level ned til f.eks WARN (warning) men da mister man en del informasjon i loggen som vil være nyttig den dagen man måtte ha behov for å feilsøke.

Plassering av loggfiler er angitt i konfig-filen under `log4net-> appender-> file`.

Mest vanlig plassering er dette: `c:\windows\temp\Cerebrum2EphorteService\Service.log`