TDialog – konfiguration

Innehållsförteckning

OBS: Konfiguration i TDialog är i grunden förlåtande, men det finns en konfigurationsinställning som är farlig och som aldrig ska användas i en TDialog-installation. Eftersom TDialog använder Hibernate för koppling mot databas kan konfigurationsparametrar från Hibernate användas i application.properties för att påverka koppling mot databas. En av dessa parametrar är spring.jpa.hibernate.ddl-auto. Denna parameter kan radera och återskapa tabeller i databasen automatiskt, vilket är farligt när databasen innehåller kunddata! Denna parameter har aldrig något relevant syfte i TDialog och *skall aldrig användas*.

1. Inledning

Nedan beskrivs de konfigurationsinställningar som finns i en installerad TDialog. För information om installationen hänvisas till TDialogs installationsdokumentation.
Vid de inställningar som visar attributnamn och värden i **fetstil** så handlar det om attributfilen config/application.properties om inget annat anges.

2. Administratörsgränssnittet – åtkomst

Endast en användare med administratörsbehörighet kan se och komma åt administrationsgränssnittet. Behörighet till administratörsgränssnittet sätts enligt 7.1.
En användare med behörighet ser en länk vid namn ”inställningar” i nedre högra hörnet i gränssnittet. Under ”inställningar” finns tre saker som kan göras: Automatrensning av meddelanden och användare, söka och ta bort användare samt krypteringslösenord. Dessa beskrivs på respektive plats i denna guide.

3. Adressering av mottagare

I TDialog finns möjlighet att ange en LDAP-källa som slår upp vilka interna adresser som finns tillgängliga. Denna källa gör dels att en intern användare får adressförslag till andra interna användare när adresser skrivs och dels att användaren inte kommer kunna skicka till en felaktig intern adress. Man har helt enkelt definierat vilka interna adresser som är tillåtna.
Varje gång TDialog startas samt en gång per dygn (se 6.2) kommer TDialog att cachea alla adresser från katalogkällan. Denna cache kommer att användas för att föreslå adresser. När man väl skickar till en intern användare och om användaren inte finns i cachen så kommer en live-uppslagning att ske mot katalogkällan.
Om LDAP-källa inte är konfigurerad så kommer inga sådana resultat komma upp som sökförslag, och då kommer användaren kunna skicka till vilka interna adresser som helst.
Sedan 3.16 kan TDialog visa inte bara en lista på e-postadresser utan även andra fält från LDAP-källan. Se server.ldap.attributes nedan.

3.1. Grundkonfiguration

server.ldap.domain=
Domänen som ska användas vid sökning

server.ldap.serviceuser=uid=td-service,ou=serviceAccounts,ou=Groups,dc=local
Serviceanvändaren (I LDAP-källan) som ska göra sökningen

server.ldap.servicepwd=dfhgfgh
Serviceanvändarens lösenord

server.ldap.ldapUrl=ldap://82.102.5.51:389
URL till LDAP-källa

server.ldap.ldapBaseDn=ou=people,dc=local
Sökbasen till LDAP-källan

3.2. Valfria parametrar

Parametrarnas defaultvärden visas i exemplet

server.cachedUserInfo.numberOfLettersForSearch=2
Med denna konfiguration så börjar sökningen efter adressförslag när användaren skrivit två tecken i e-postadressen. Observera att förslagen inte kommer upp förrän användaren skrivit 3 tecken.

server.ldap.objectClassExpression=(objectClass=user)
Hur sökningen efter användare går till. Om LDAP-källan inte är ett AD så kan denna behöva förändras.

server.ldap.ldapSearchChunk=1000
Med denna inställning tar TDialog ut 1000 adresser i taget ur LDAP-källan, för att sedan göra en ny sökning.

server.ldap.allowSendToUnknownInternalAddresses=false
Om attributet är false kommer TDialog inte tillåta att skicka till en intern användare som inte finns i adresslistan (default=true).

server.ldap.attributes=mail,department
Per default visas e-postadress (mail) i resultatlistan vid LDAP-sökning. Men, sedan 3.16 kan även andra attribut visas genom att ange en lista i denna inställning. Attributen visas i ordningsföljd, så listan blir en ”tabell” när de visas. Notera att det finns begränsningar i storlek, framför allt vid användning på mindre devices, för vilken info som är lämplig att visa. **OBS:** Det första attributet i listan måste alltid vara det där e-postadressen finns (vanligtvis mail).

3.3. startTLS

Från och med TDialog 3.9 finns möjlighet att köra LDAP-anslutningen för adresslistor via startTLS. För att göra detta krävs följande konfiguration:

server.ldap.trustStore

Det trustStore där certifikatet för LDAP finns. Det måste vara en JKS. Se nedan hur en sådan kan skapas.

server.ldap.trustStorePwd
Lösenord för trust store. Se nedan hur en sådan kan skapas.

Att skapa trustStore för LDAP-certifikat:

- `openssl s_client -connect ldaphost.se:636` (kräver att 636 är åtkomlig)
- Detta kommando ger ett certifikat tillbaka i base64-format. Spara ner detta som en fil vid namn ldaps.crt.
- Skapa sedan en tom JKS (dvs skapa JKS med alias och ta sedan bort aliaset).
  ```
  keytool -genkey -keyalg RSA -alias tempalias -keystore truststore.jks
  keytool -delete -alias tempalias -keystore truststore.jks
  ```
- Lägg sedan till ditt certifikat:
  ```
  keytool -import -v -trustcacerts -alias ldaps -file ldaps.crt -keystore truststore.jks
  ```
- Sedan innehåller trustStore.jks det certifikat du behöver lita på.

4. Attributkonfiguration

TDialog konsumerar attribut från en SAML-IDP. Dessa attribut används för att bestämma vem användaren är (inloggningsnamn, riktigt namn, e-postadress) och de används även för behörigheter till administratörsgränssnitt och funktionsbrevlådor. Att dessa attribut är konfigurerade korrekt är en förutsättning för att användaren ska kunna logga in.

4.1. Personattribut

server.auth.fullUserAttribute.attributeName=urn:oid:2.5.4.11
server.auth.fullUserAttribute.acceptedValues=staff
Endast en användare med server.auth.fullUserAttribute.attributeName satt till server.auth.fullUserAttribute.attributeName kommer att få åtkomst till TDialog. Syftet med attributet är att kunna begränsa åtkomst till bara vissa av de som kan logga in på IDP:n.

server.auth.userAttribute=urn:oid:0.9.2342.19200300.100.1.1;Subject_SerialNumber
userAttribut är attributet som sätter användarnamnet, dvs TDialogs interna identifierare på personen. För en intern inloggning är det klokt att använda sAMAccountName eller UPN i katalogen. För en extern inloggning är personnummer vanligt, eftersom den externa inloggningen ofta utförs med BankID.

server.auth.emailAddressAttribute=urn:oid:0.9.2342.19200300.100.1.3
Användarens e-postadress. Denna anges av avsändare för att skicka meddelanden till användaren, och notifieringar skickas med e-post till denna e-postadress.

server.auth.displayNameAttribute=urn:oid:2.16.840.1.113730.3.1.241

Denna används för användarens namn. Attributet används endast visuellt, men det är ett obligatoriskt attribut.

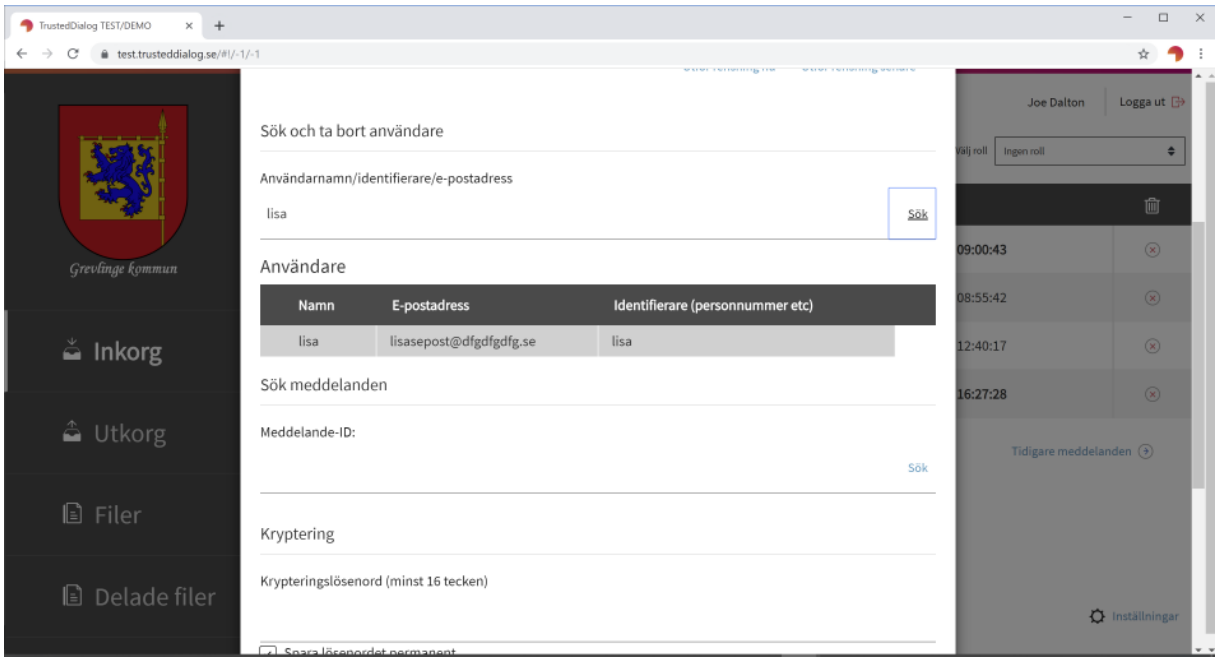
4.2. Behörighetsgivande attribut

server.auth.adminAttribute=urn:oid:1.3.6.1.4.1.5923.1.8
server.auth.adminAttributeValue=admin|superAdmin
Dessa beskrivs närmare under avsnittet “behörigheter”.

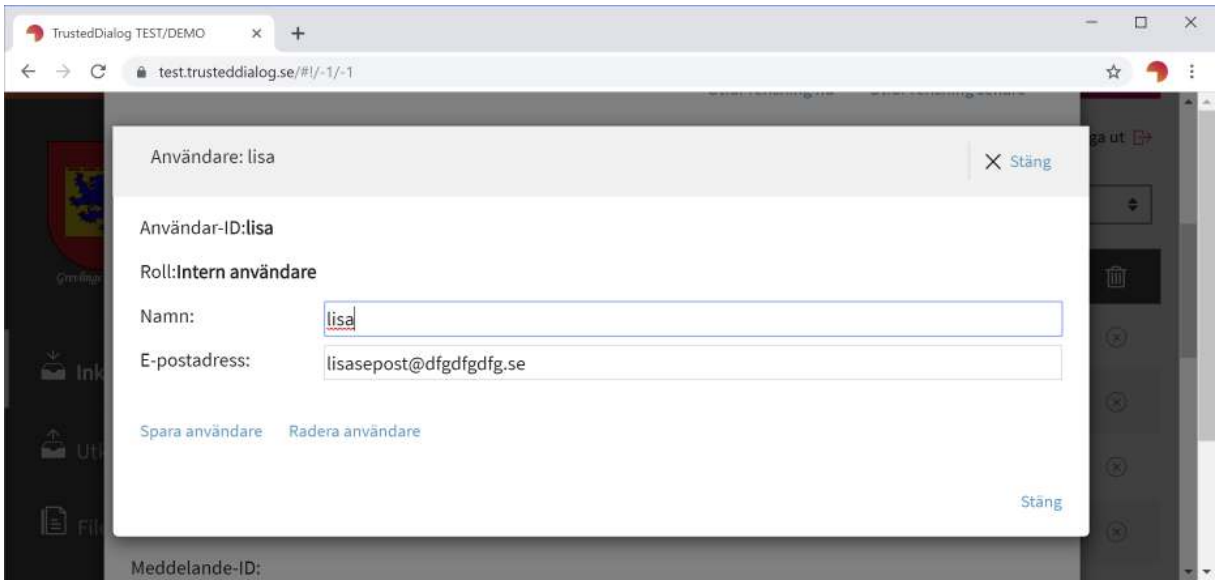
5. Användare, redigera och radera

I administrationsgränssnittet kan en administratör radera, och sedan 3.6 även redigera, användare. Namn och e-postadress kan förändras på alla användare. Identifierare kan aldrig förändras, eftersom detta är en del av autentiseringen. (Om en administratör kunde ändra identifierare så skulle den kunna ta del av samtliga användares meddelanden, genom att ändra identifierare till en identifierare som administratören kan logga in som).
Gör följande för att redigera eller radera en användare:

1. Logga in som en användare med administratörsbehörighet
2. Klicka på inställningar längst ner till höger
3. Sök rätt på den/de användare som ska redigeras/raderas



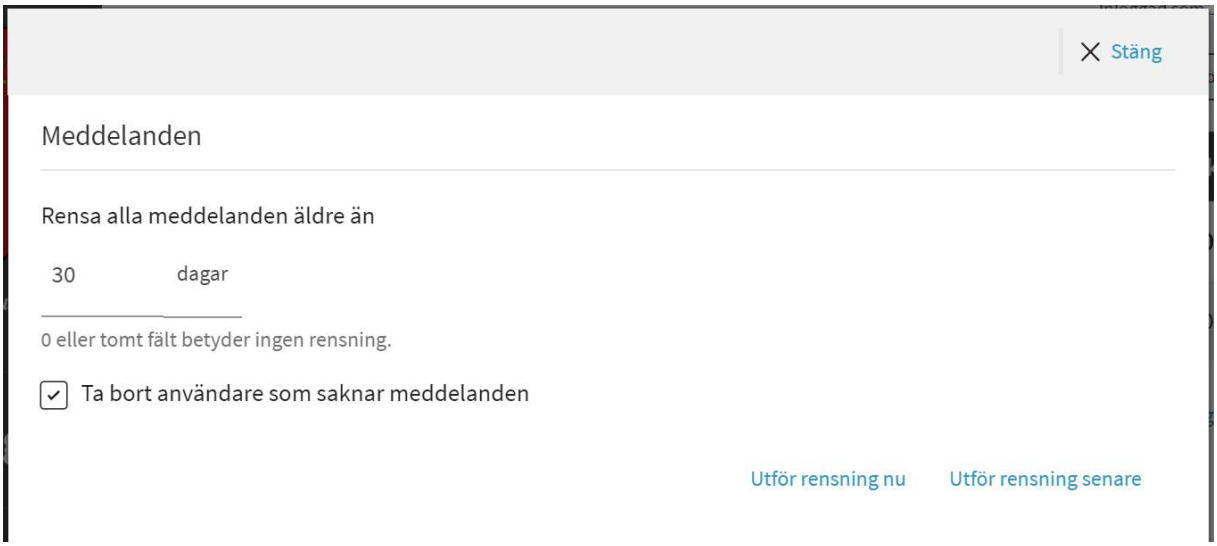
4. Klicka på användaren



5. För att radera, klicka på ”radera användare” och därefter OK. För att redigera, ändra i de ändringsbara fälten och tryck på ”spara”.

6. Automatisk rensning av meddelanden och användare

6.1. Gränssnittsdelen



I administratörsgränssnittet (se 2) finns inställningar för hur automatrensning av meddelanden ska gå till. Man kan ange följande inställningar:

Rensa alla meddelanden äldre än X dagar. Om man anger mer än 0 dagar så kommer TDialog enligt intervallet som anges i 6.2 att radera alla meddelanden som är äldre än det som anges här. Notera att TDialog inte tar hänsyn till konversationer utan behandlar alla meddelanden individuellt. Ett svar på ett meddelande kan alltså finnas kvar medan ursprungsmeddelandet raderas. Precis som med e-post!

Ta bort användare som saknar meddelanden. Om denna inställning väljs så kommer TDialog efter rensning av meddelanden att kolla om det finns ”tomma” användare, dvs användare utan meddelanden. Om så är fallet kommer dessa användare att raderas. Notera att TDialog sparar

väldigt lite information om användare, det mesta sparas i katalogtjänsten (AD:t). Därför är det tämligen ofarligt att ta bort användare, och interna användare kommer att läggas till med automatik igen när de loggar in på nytt. Den enda märkbara skillnaden blir på externa användare som loggat in med e-legitimation och där personnummer knutits till en e-postadress, för om man skriver till en sådan användare som blivit borttagen så måste man ange personnumret på nytt.

När man valt inställningar klickas **schemalägg rensning** för att spara och komma vidare.

6.2. Automatjobbsinställningar

server.jobs.cronParam=0 0 2 * * *

Inställningen bestämmer när rensningsjobb ska köras för att eventuellt ta bort gamla användare och meddelanden (om man inte angett att meddelanden ska rensas så har inställningen ingen effekt). Inställningen är i cron-format (http://www.quartz-scheduler.org/documentation/quartz-2.x/tutorials/crontrigger.html) och exemplet ovan innebär att rensningsjobb utförs kl 02:00 varje natt. Det är också defaultbeteendet.

6.3 Rensa kvarvarande bilagor

Ibland misslyckas TDialog med att ta bort bilagor i samband med att ett meddelande tas bort. För att hantera det finns ett väldigt grundläggande gränssnitt där man kan se om det finns sådana kvarvarande bilagor och även rensa dem.

Arbetsgång för att rensa kvarvarande bilagor:

- Ta en backup på databas och fillager
- Logga in som en administratör
- Gå till följande länk: **https://<host>/admin/findOrphans**
- Här syns vilka kvarvarande bilagor som finns
- För att rensa kvarvarande bilagor, gå till: **https://<host>/admin/findOrphans?deleteOrphans=true**
- Bilagorna raderas och status kommer på skärmen

7. Behörigheter

7.1. Administratör

server.auth.adminAttribute=urn:oid:1.3.6.1.4.1.5923.1.8

server.auth.adminAttributeValue=admin|superAdmin

Inställningen server.auth.adminAttribute bestämmer vilket attribute som läses för att ta reda på om en användare är administratör i TDialog. Inställningen server.auth.adminAttributeValue beskriver det värde som server.auth.adminAttribute ska ha för att användaren ska vara administratör. server.auth.adminAttributeValue är ett reguljärt uttryck (regular expression https://en.wikipedia.org/wiki/Regular_expression), dvs admin|superAdmin betyder att attributvärdet kan vara admin ELLER superAdmin, dvs exemplet betyder att en admin måste ha attributuppsättningen:

urn:oid:1.3.6.1.4.1.5923.1.8=admin

eller

urn:oid:1.3.6.1.4.1.5923.1.8=superAdmin

för att anses vara administratör i TDialog.

7.2. Funktionsbrevlådor

server.auth.userRolesAttribute=urn:oid:1.3.6.1.4.1.5923.1.1.1.7
Inställningen bestämmer vilket SAML-attribut som anger funktionsbrevlådor. Enligt inställningen ovan så skulle exempelvis attributet
urn:oid:1.3.6.1.4.1.5923.1.1.1.7=socialtjanst@grevlinge.se;ungdomsmottagning@grevlinge.se
Betyda att användaren har åtkomst till funktionsbrevlådorna socialtjanst@grevlinge.se och ungdomsmottagning@grevlinge.se.

7.3. Externa användare

server.guestUser.permission=70
Inställningen bestämmer vad en gästanvändare kan göra i systemet. De möjliga värdena är:

10: Gästanvändare kan bara besvara meddelanden, inte skicka till nya användare.
50: Gästanvändare måste bjudas in, men kan skicka meddelanden till vem de vill efter inbjudan
70: Gästanvändare kan självregistrera, och skicka meddelanden till vem de vill efter inbjudan.

server.guestUser.privilegedAddresses=@grevlinge.se;lisa@annandoman.se
I en installation där guestUser.permission är 10 kan man ange adresser som ska ha motsvrande 50 som behörighet, dvs kunna starta nya konversationer eller vidarebefordra meddelanden. Avskiljare är som vanligt semikolon, och notera att notationen tillåter en att ange både enskilda adresser och hela e-postdomäner.

8. Certifikatsinställningar

server.ssl.key-store=/trusteddialog/config/https-cert.pfx
server.ssl.key-store-password=Vinter15
Inställningarna styr över det certifikat som används för att skapa https-trafik till TDialog. server.ssl.key-store anger var certifikatsfilen befinner sig (p12-format) och server.ssl.key-store-password anger lösenordet till p12-filen.

9. Databasanslutning

MySQL

spring.datasource.url=jdbc:mysql://localhost/trusteddialog?useSSL=false&autoReconnect=true
spring.datasource.username=tdialoguser
spring.datasource.password=Vinter15

Inställningarna anger hur TDialog kopplar upp sig mot databasen. Username och password anges automatiskt vid installation, men kan ändras om man vill skapa ett annat servicekonto i databasen.

MS SQL Server

Sedan 3.6 finns även möjlighet att använda MS SQL Server. Observera att man då använder en befintlig SQL Server-instans och att den inte är en del av installationen. Notera även att installationsscriptet gör en MySQL-konfiguration, så inställningarna i application-prod.properties måste ändras manuellt för en MS SQL Server.

spring.datasource.url=jdbc:sqlserver://dbserver:1433;databaseName=trusteddialog
spring.datasource.username=tdialoguser
spring.datasource.password=Vinter15
spring.datasource.driver-class-name=com.microsoft.sqlserver.jdbc.SQLServerDriver

OBS: I SQLServer behöver man på något sätt också ange vilket databasschema man ska använda när man ansluter. TDialogs schema heter trusteddialog. Detta kan konfigureras mot SQLServer på två sätt:

1. Man sätter default schema=trusteddialog på användaren i SQL server.
2. Man sätter **spring.jpa.properties.hibernate.default_schema=trusteddialog** i application.properties.

10. Elektroniska underskrifter

Notera att det finns ett separat dokument som mer grundligt beskriver konfiguration av PhenixID Signing.

Från och med TDialog 3.2 finns en integration med PhenixID Signing för att elektroniskt skriva under PDF-dokument. Om integrationen är påslagen och användaren är på väg att skicka ett eller flera PDF-dokument så kommer det dyka upp en förfrågan om att skriva under, och om användaren svarar ja så kommer den skickas vidare till PhenixID för själva underskriften.

Följande inställningar i application-prod.properties konfigurerar elektroniska underskrifter:

server.signing.phenixid.enabled=true
Slår på integrationen, dvs det kommer komma en fråga när användaren försöker skicka ett PDF-dokument

server.signing.phenixid.private-key-file= /private_key_file.key
I samband med integrationen skapas ett nyckelpar för trust mellan TDialog och PhenixID. Inställningen ovan är pathen till nyckelparets privata nyckel.

server.signing.phenixid.signing-domain=signing.trusteddialog.se:8443
Signing-serverns hostnamn och port

server.signing.phenixid.jwt-recipient-path=/td
server.signing.phenixid.rest-signing-path=/files/integration_dev

server.signing.phenixid.web-signing-path=/sign
Ovanstående inställningar är interna inställningar för PhenixID, som sällan eller aldrig behöver modifieras.

11. Engångsmeddelanden

server.onetime.enabled=true
Aktiverar engångsmeddelanden

server.onetime.maxPwdAttempts=5
Antalet försök som tillåts att slå in rätt siffror från SMS:et

server.onetime.smsHttpMethod=PUT
server.onetime.smsHttpUrl=https://smsgate.example.se/sendSMS
server.onetime.smsHttpHeaders={Content-Type: application/json, tenant: t1}
server.onetime.smsHttpPayload={"mobile":"$smsNumber$","message":"otp"}
server.onetime.smsServiceClientCert=c:/trusteddialog/config/sms_service_client.p12
server.onetime.smsServicePwd=
Konfigurationsinformation till SMS-tjänsten. Ovan ett exempel från PhenixID:s SMS-gateway

server.onetime.oneTimeRecipientLifespan=5
Livslängden i dagar för engångsmeddelanden.

12. E-postintegration

I TDialog 3.11 finns möjlighet att både skicka och ta emot krypterade e-postmeddelanden. Funktionen kallas tekniskt för mailconnector.

På grund av certifikat mm har mailconnector ganska komplicerad konfiguration, som hänger tätt ihop med konfiguration av e-postserver och hur avsändarens e-postmeddelanden ser ut. TDialog rekommenderar att kunden låter sin supportpartner göra den initiala konfigurationen. Nedan kommer en beskrivning av konfigurationsparametrarna.

Mailconnector.active=true
Aktiverar själva funktionen

Mottagning av krypterade e-postmeddelanden

mailconnector.receive.cronParam=*/10 * * * * *
Det intervall man önskar leta efter inkommande e-postmeddelanden. Standard cron-format, exemplet visar var tionde sekund

mailconnector.receive.imapHost=grevlinge.se
Mottagande mejlserver.

mailconnector.receive.imapPort=143
Mottagande mejlserver-port

```
mailconnector.receive.allowedRecipients=[{"name":
"genericmobile@grevlinge.se", "emailPassword": "Vinter15",
"internalRecipient": "kundtjanst@grevlinge.se"}]
```

Vilka konton (e-postadresser) i TDialog är tillåtna att ta emot e-postmeddelanden? ”Name” är e-postadressen till kontot i e-postservern, ”emailPassword” är lösenordet till e-postservern och ”internalRecipient” är det konto (e-postadress) i TDialog som TDialog-meddelanden ska skickas till. Det blir inte bra om name och internalRecipient är samma, eftersom TDialog skickar ett eget notifieringsmeddelande som då hamnar i e-postlådan tillsammans med de krypterade e-postmeddelandena.

```
mailconnector.receive.senderDecryptionStrategies=[{"id": "filePgpId",
"type": "attachmentPgp", "keyPassword": "Vinter15", "privateKeyPath":
"/var/lib/trusteddialog/config/mailconnector-
credentials/jonastjanst_private.asc"}]
```

Hur TDialog förväntar sig att inkommande e-postmeddelanden krypteras. TDialog har stöd för PGP och, i begränsad utsträckning, S/MIME. För PGP, använd ID och typ enligt exempel och sätt sökväg och lösenord.

```
mailconnector.receive.allowedSenders=[{"name":
"+46103334566@fax.genericmobile.se", "allowedRecipients":
["genericmobile@grevlinge.se"], "cryptoStrategy": "filePgpId",
"certPath": "/var/lib/trusteddialog/config/mailconnector-
credentials/generic_public.asc"}]
```

Vilka avsändare är tillåtna att skicka till respektive allowedRecipient? Avsändaren har:

- Ett namn
- En lista på tillåtna mottagare (som måste vara definierade i allowedRecipients)
- En cryptoStrategy (som måste vara definierad i senderDecryptionStrategies)
- En certPath (certifikatet som används för verifiering och dekryptering av avsändarens inofrmation)

Skickande av krypterade meddelanden

```
mailconnector.send.smtpHost=localhost
mailconnector.send.smtpPort=25
```

Host och port till avsändande e-postserver.

```
mailconnector.send.recipientEncryptionStrategies=[{"id":
"genericMobilePgp", "type": "attachmentPgp", "encryptionPublicKeyPath":
"/var/lib/trusteddialog/config/mailconnector-
credentials/generic_public.asc"}]
```

EncryptionStrategy för avsändaren, dvs den nyckel som krävs för att kryptera meddelanden.

```
mailconnector.send.senderSigningStrategies=[{"type": "attachmentPgp",
"privateKeyPath":"/var/lib/trusteddialog/config/mailconnector-
credentials/jonastjanst_private.asc", "privateKeyPwd": "Vinter15"}]
```

SigningStrategy för avsändaren, dvs den nyckel och det lösenord som behövs för att signera meddelanden.

mailconnector.send.allowedRecipients=[{"name": "jonasstj@gmail.com", "recipientCryptoStrategy": "genericMobilePgp", "signingStrategyType":"attachmentPgp"}]

Vilka mottagare är tillåtna att skicka till från TDialog? E-postadress och recipientCryptoStrategy (som definieras i recipientEncryptionStrategies). signingStrategyType är alltid attachmentPgp för PGP över e-post.

mailconnector.send.allowedSenders=[{"name": "genericmobile@grevlinge.se", "tdSender": "kundtjanst@grevlinge.se", "smtpPassword": "Vinter15", "allowedRecipients":["jonasstj@gmail.com", "+46103334566@fax.genericmobile.se"]}]

Vilka TDialog-användare får skicka krypterade e-postmeddelanden, och till vem? Name är e-postadressen i e-postservern, tdSender är TDialog-användaren (e-postadress), smtpPassword är lösenordet som eventuellt behövs för att skicka via e-postservern, allowedRecipients behöver vara definierade i mailconnector.send.allowedRecipients.

13. Extern html, css etc

server.frontend.externalFilePath=/trusteddialog/html/

Inställningen bestämmer var de externa HTML-filerna befinner sig. De externa HTML-filerna är logotyper, CSS:er etc.

14. Filuppladdning

spring.servlet.multipart.max-file-size=10MB

spring.servlet.multipart.max-request-size=10MB

Inställningarna bestämmer maxstorleken på uppladdade filer. Filstorleken är själva filen och requeststorleken är hela requestet. För megabytestora filer är skillnaden försumbar, men båda inställningarna behövs i det underliggande ramverket

15. Gränssnittskonfigurationer

server.gui.showFilesView=true

TDialog har en filuppladdningsfunktion, där användaren kan ladda upp filer till sig själv (ej att förväxla med bifogade filer i meddelanden som är något helt annat). Inställningen bestämmer helt enkelt om denna filvy ska visas för användaren eller ej.

server.gui.logoFileName=logo-white.svg

server.gui.logoLandscapeFileName=logo-white-landscape.svg

Filnamn för de logotyper som visas av TDialog. Grundstorleken på filerna är 164 bredd och 67 i höjd för logoFileName och 443 bredd och 64 höjd. Men designen anpassar sig även till andra bildstorlekar.

server.gui.inboxName=Inkorg

server.gui.outboxName=Utkorg

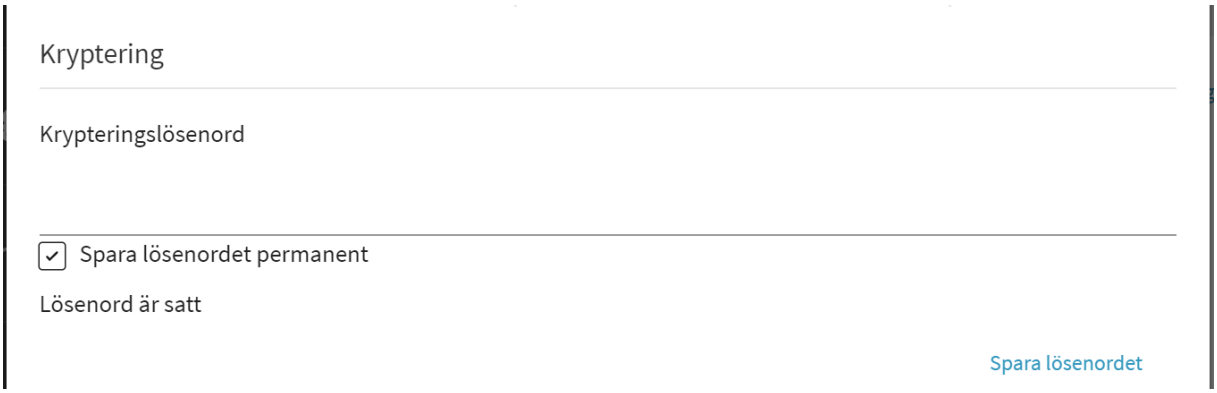
server.gui.filesName=Filer

Dessa inställningar kan ändra namnen på “Inkorg”, “Utkorg” och “Filer” i gränssnittet.

server.gui.helpPageUrl=https://www.grevlinge.se/trusteddialog/help
Denna inställning anger en URL-för en hjälpsida. Om inställningen finns så dyker det upp en hjälpknapp som länkar till URL:en. Det innebär att man som kund kan skapa sina egna hjälpsidor.

server.gui.conversationView=true
Denna inställning anger default för användarna vad gäller konversationsvyn (true betyder konversationsvy, false betyder meddelandevy). Användaren kan själv alltid ändra vy och om TDialog kommer ihåg användarens preferens. Denna inställning är bara för användare som inte aktivt ändrat. Om detta inte är satt kommer användarna att som default få meddelandevyn.

16. Kryptering



TDialog har stöd för kryptering av rubriker, meddelandetexter och filer. Observera att själva filnamnet inte blir krypterat, men allt annat potentiellt känsligt innehåll krypteras.
Lösenordet för kryptering kan sättas på två sätt:

- Lösenordet sparas permanent. Lösenordet lagras krypterat på TDialog-servern.
- Lösenordet sparas inte permanent. Det innebär att lösenordet måste anges av en administratör vid varje uppstart av TDialog för att någon ska kunna läsa meddelanden och filer. Denna inställning skyddar TDialogs innehåll ännu mer än om krypteringslösenordet lagras på servern, men i gengäld kommer TDialog inte att kunna användas om servern startar om sig mitt i natten innan en administratör gått in och angett lösenordet.

Tanken är att lösenordet aldrig ändras, eftersom det används för att kryptera informationen. Informationen är inte läsbar utan lösenordet och det finns i dagsläget ingen funktion för att ”konvertera” informationen till ett nytt lösenord. När man försöker ange ett nytt lösenord så testar TDialog om det nya lösenordet kan låsa upp informationen. Om lösenordet inte stämmer så sätts lösenordet inte om och det gamla bevaras (om ett lösenord är sparat).

OBS: Lösenordet måste vara minst 20 tecken långt och rekommenderas ha hög entropi. Detta för att även någon som stulit innehållet i databasen ska ha rejäla problem att knäcka det. Förslagsvis används en lösenordshanterare med stark autentisering. Observera vidare att lösenordet krävs för att få tillgång till TDialogs information, även av den ”rätta ägaren till informationen”. Det finns

ingen bakdörr för leverantör, supportpersonal eller liknande utan man behöver tillgång till det lösenordet för att kunna läsa informationen.

17. Länkar in i TDialog

Förutom att skapa länkar till den externa och interna inloggningen är det även möjligt att skapa länkar som påbörjar skapande av specifika meddelanden. I Outlook-pluginen används funktionen exempelvis för att direkt komma in i TDialog med en mottagaradress och en meddelanderubrik. På samma sätt kan man skapa länkar exempelvis på sin webb för att påbörja ett specifikt ärende. För att ange exempelvis att vi efter inloggning ska öppna ett meddelande till bygglov@grevlinge.se med rubrik ”Bygglovsansökan” i vår testmiljö anger vi exempelvis följande länk:

https://test.trusteddialog.se/index.html?toAddress=YnlnZ2xvdkBncmV2bGluZ2Uuc2U&openNewMessage=true&messageTitle=QnlnZ2xvdnNhbnPDtmthbg

toAddress och messageTitle är inte så krypterade som det ser ut, de är bara omkodade med base64-encoding för att kunna skickas med i en URL på detta sätt. (Använd exempelvis https://www.base64decode.org/ för att göra base64-encodningar av icke-känsligt material.)

Problemet med exemplet ovan är att det blir en intern inloggning, och för exemplet bygglov och de flesta andra liknande exempel vill man ju ha en extern inloggning. Man vill helt enkelt ha en enkel ”e-tjänst” för invånare där de skickar ett meddelande automatiskt adresserat till rätt person.

Från TDialog 3.9 är detta möjligt att göra, enligt nedan exempel:

https://test.trusteddialog.se/index.html?localEntityId=https://sp-guest.trusteddialog.se&toAddress=YnlnZ2xvdkBncmV2bGluZ2Uuc2U&openNewMessage=true&messageTitle=QnlnZ2xvdnNhbnPDtmthbg

Observera den tillagda localEntityId-parametern, som är entity-ID för guestUser-användare i TDialog.

Er installationspartner hjälper er gärna att skapa dessa länkar.

18. Mina meddelanden

server.minameddelanden.enabled = true
Aktiverar mina meddelanden-kopplingen

OBS: Sedan 3.21 används **server.minameddelanden.enabled** inte längre utan istället är det

server.minameddelanden.enabledAttribute
server.minameddelanden.enabledAttributeValue

som gäller. Om man exempelvis vill att bara de användare som har attributet ”minameddelandenadmin” satt till ”true” eller ”1” ska kunna skicka så blir inställningen:

server.minameddelanden.enabledAttribute=minameddelandenadmin
server.minameddelanden.enabledAttributeValue=true|1

Vill man istället att alla ska kunna skicka Mina meddelanden gör man:
server.minameddelanden.enabledAttribute=urn:oid:0.9.2342.19200300.100.1.1
server.minameddelanden.enabledAttributeValue=.*
(Give tatt användarnamnen finns på attribut urn:oid:0.9.2342.19200300.100.1.1). Man säger då: ”Alla som har ett användarnamn ska kunna skicka Mina meddelanden”.

server.minameddelanden.file-types=.pdf;.docx
De filtyper som önskas tillåtas för bilagor till Mina meddelanden.

server.minameddelanden.endpoint_url = https://www5.skatteverket.se/webservice/ec
URL till Mina meddelanden i produktion

server.minameddelanden.ssl_java_keystore_path = c:/trusteddialog/config/mm-credentials/trust.jks
server.minameddelanden.ssl_java_keystore_password =
Keystore som innehåller https-certifikatet till Mina meddelandens webbtjänst.
server.minameddelanden.ssl_sender_pkcs12_path = /var/lib/trusteddialog/config/mm-credentials/server-certifikat-5c1217796deb9.p12
server.minameddelanden.ssl_sender_pkcs12_password = 3755318131025899
Server-certifikat för Mina meddelanden

server.minameddelanden.sender_pkcs8_cert_path = /var/lib/trusteddialog/config/mm-credentials/td-stampel.crt
server.minameddelanden.sender_pkcs8_key_path = /var/lib/trusteddialog/config/mm-credentials/td-stampel.p8
Stämpelcretifikat för Mina meddelanden

server.minameddelanden.sender_orgnr = 165591248447
server.minameddelanden.sender_name = TDialog
Avsändarens organisationsnamn och organisationsnummer.

server.minameddelanden.header.language = svSE

server.minameddelanden.support.email = enadress@organisationen.se
server.minameddelanden.support.phone = +46 70 1234566
server.minameddelanden.support.url = https://en.supportadress.nu

server.minameddelanden.support.text = Support
Supportinformation som visas för mottagaren av meddelandet i Kivra/Min myndighetspost.

server.minameddelanden.email.reply_to = test-mm.svara.ej@trusteddialog.se
server.minameddelanden.email.subject = Min myndighetspost - avisering från TDialog
server.minameddelanden.email.text = Du har nya meddelanden från TDialog, https://www.minmyndighetspost.se/
Text som visas i notifieringsmeddelandet från Kivra/Min myndighetspost

19. Notifieringsinställningar

19.1. E-postgateway

server.mail.host=smtp.grevlinge.se
server.mail.username=Administrator
server.mail.pwd=Vinter15
Dessa inställningar hanterar TDialogs e-postnotifieringar. server.mail.host är hostnamnet till e-postgateway och server.mail.username respektive server.mail.pwd är inloggningsuppgifter till e-postgateway om sådana krävs.

server.mail.senderEmail=trusteddialog@trusteddialog.se
Inställningen bestämmer avsändarens e-postadress på notifieringarna i TDialog.

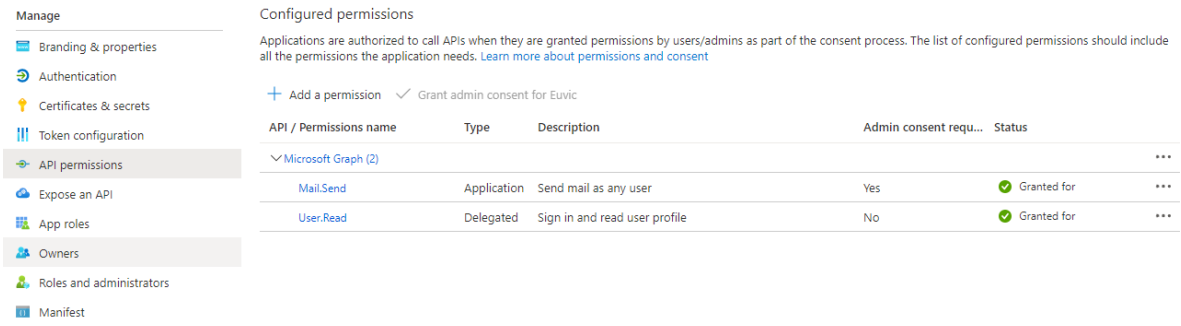
StartTLS (exempelvis Office365)
E-postgatewayen kan även konfigureras med startTLS, enligt följande. Exemplet är från Office365
server.mail.host=smtp.office365.com
server.mail.port=587
server.mail.username=<adressen som ska användas för inloggning>
server.mail.pwd=<inloggningskontots lösenord>
server.mail.senderEmail=<frånadressen. I fallet med Office365 måste den vara identisk med server.mail.username>
server.mail.trustStoreFile=<jks innehållande gatwayens issuing CA>
server.mail.trustStorePwd=<lösenord till ovan jks>

19.2. Skicka e-notifieringar med Microsofts Graph-API

Nya Best Practice från Microsofts hall är att skicka e-postmeddelanden med hjälp av deras Graph API. Man använder då protokollet OAuth för att kvittera ut en token och sedan använda den för att skicka ett e-postmeddelande. Man använder då inte protokollet SMTP som traditionellt används för e-post, utan skickar istället ett REST webservicemeddelande till MS Graph som skickar själva e-postmeddelandet.

Behörigheter

Följande behörigheter behövs för det servicekonto som ska kvittera ut en token och skicka ett e-postmeddelande i MS Graph:



Konfiguration

 # adress som blir avsändaradress, och som är kontot som loggar in i e-posttjänsten i MSGraph
server.msgraphemail.fromAddress=noreply@grevlinge.se

 # Följande tre måste man få från Microsoft. Dessa är endast exempel.
server.msgraphemail.tenant=b92ff8fb-c01f-4ac6-a166-e6516277cdc8
server.msgraphemail.clientId=1a64c4bd-3a23-478e-a516-acc71c62070f
server.msgraphemail.clientSecret=afcaa482-18ee-4f19-bec3-5657d5c00209

 # truststore där Microsofts alla utfärdade finns. Förslagsvis används Javans cacerts-fil
server.msgraphemail.msOnlineTrustStorePath=/usr/lib/jvm/zulu-11-amd64/lib/security/cacerts

 # Standardlösenordet för Javans cacerts-fil
server.msgraphemail.msOnlineTrustStorePwd=changeit

19.3. Notifieringsmeddelandenas texter

Inledning

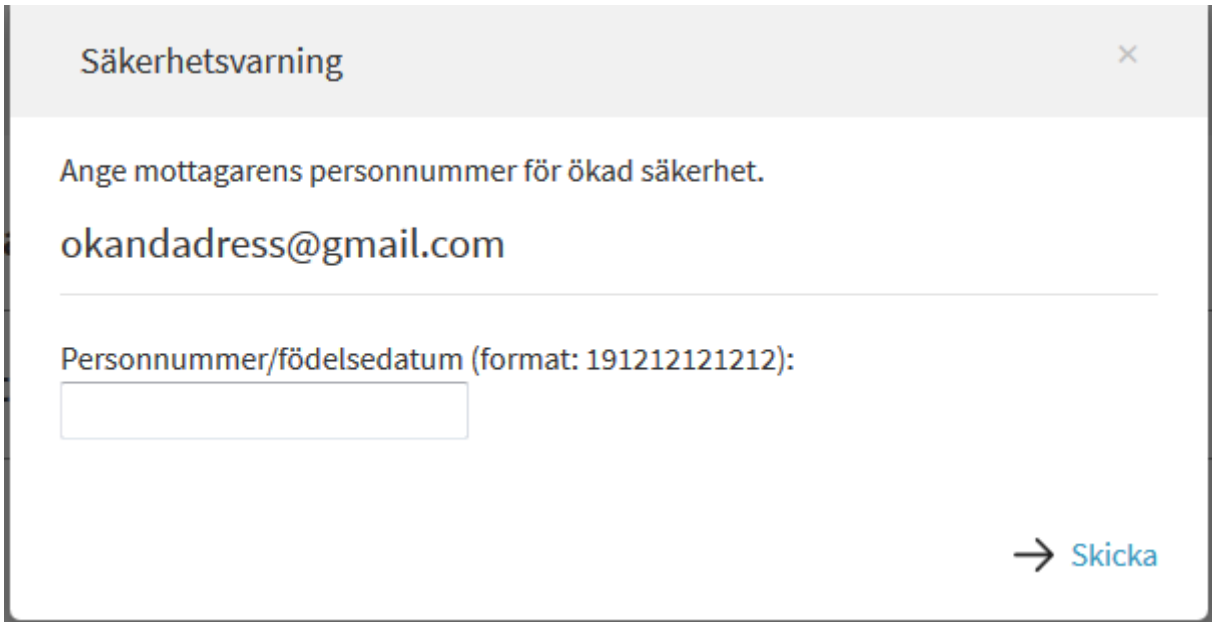
De notifieringsmeddelanden som sänds kan konfigureras helt med inställningar i filen **config/application-prod.yml**. Notera alltså att dessa finns i en separat konfigurationsfil. Anledningen är att detta filformat hanterar svenska tecken, radbrytningar osv som är nödvändiga för att kunna konfigurera texter i e-postmeddelanden.
Formatet i config/application-prod.yml fungerar så att hierarkier av inställningar bestäms av positionen i filen. Man kan säga att

server:
auth:

motsvarar

server.auth

I application-prod.properties. För att kunna ange attributen på ett rimligt sätt i detta dokument så anges application-prod.yml-inställningar som:
server: auth:
dvs radbrytningen och indraget tas inte med. Detta av läsbarhetsskäl.



server: securityWarning: fieldText: Personnummer
Texten på personnummerfältet på bilden ovan

server: securityWarning: html: "Skriv in mottagarens personnummer i formatet ÅÅÅÅMMDDNNNN (12 siffror)"
Brödtexten i bilden ovan.

unknownAddress: html: "Verifiera att e-postadressen är korrekt"
Varningsmeddelande om användaren försöker skriva till en okänd adress.

mail: notificationSubject: "Du har fått ett säkert meddelande från Grevlinge kommun"
Rubrik för e-postmeddelande till interna användare.

mail: notificationSubjectGuestUser: "Du har fått ett säkert meddelande från Grevlinge kommun"
Rubrik för e-postmeddelande för redan registrerade gästanvändare

mail: notificationBodyPlainText: "Hej $displayName$, Klicka här: https://www.grevlinge.se/messageId/$messageId$/"
Meddelandetext för interna användare. Observera användandet av $displayName$ och $messageId$. Dessa fylls i med korrekt mottagarnamn och meddelande-ID.

mail: notificationBodyPlainTextGuestUser:
Samma som *mail: notificationBodyPlainText:* men för redan registrerade gästanvändare.

mail: otpSubject:
Samma som mail: notificationSubjectGuestUser: men för gästanvändare som ännu inte registrerats.

mail: otpBodyPlainText: ”Du har fått ett Säkert meddelande från Grevlinge kommun. Klicka på länken nedan för att läsa meddelandet.
https://@@hostname@@/connectUser/alias/guestUser?guestUser=true&guidString=otp&emailAddress=$emailAddress$&messageId=$messageId$"
Samma som mail: notificationBodyPlainText: men för gästanvändare som ännu inte registrerats. Observera variablerna $emailAddress@, otp och $messageId$.

mail: confirmationSubject:
Rubrik för e-postverifieringen för användare som självregistrerar.

mail: confirmationBodyPlainText: "Klicka på länken nedan för att verifiera din e-postadress och börja använda TDialog:
https://sakrameddelanden.grevlinge.se/connectUser/alias/guestUser?guestUser=true&guidString=otp&emailAddress=$emailAddress$&messageId=$messageId$"
Observera variablerna otp och $messageId$ i länken. Dessa ersätts med en OTP och ett meddelande-ID när notifieringen skickas.

19.4. Notifiering vid borttagande av meddelande

Normalt skickas notifieringar när meddelanden skickats men det är även möjligt att skicka notifieringar när meddelanden återtagits, dvs när avsändaren bestämt sig för att ett meddelande inte längre ska vara tillgängligt för mottagaren. För att skicka notifieringar i det fallet krävs följande konfiguration:
server.gui.revokeMessage.notify: Om denna är true aktiveras notifieringar av borttagande av meddelande.
server.mail.revokeMessageSubject: E-postrubrik i notifieringen
server.mail.revokeMessageBody: E-posttext i notifieringen

20. Office365-AddIn

Från och med TDialog 3.7 finns integration med Office365, och denna integration är inbyggd direkt i TDialog-programvaran (för Outlook 2016 och tidigare är integrationen en separat programvara).
Gör följande för att installera TDialog-addin för Office365:
Lägg till följande i application-prod.properties:
server.outlook.enabled=true
Användare som ska ladda ner addin går till:
https://sakrameddelanden.kundnamn.se/public/outlook/outlook-manifest

OBS: Ladda ner manifest-filen först till disk, använd sedan alternativet ”Add a custom add-in -> Add from file” och peka på filen.

För mer allmän information, se separat dokument om Outlook-pluginen.

20.1. Parametrisering och flera add-ins för samma installation

Eftersom Office365-addinen är knuten till TDialog blir det en liten utmaning om man önskar att olika användargrupper har olika plugins, exempelvis för att man vill att olika användargrupper använder olika IDP:er. Från TDialog 3.14 är det dock möjligt att lösa.
När användaren (eller med automatik) laddar ner pluginen används istället en specialdesignad länk, exempelvis:
https://test.trusteddialog.se/public/outlook/param-outlook-manifest?paramB64=aWRwPWh0dHBzOi8vd3d3LmV4YW1wbGUuY29tL2lkcA%3D%3D&id=02b9395e-12dd-453f-bff2-402745c454f9
Denna länk styr två saker:

1. Vilket ID som används för add-in:en (id-parametern i länken). Varje add-in i Outlook har ett ID, det är en GUID och två add-ins kan inte ha samma. Använd exempelvis ID:t ovan och byt några bokstäver/siffror för att få ett unikt ID.
2. Vilka extra parametrar som ska finnas i den resulterade Outlook-länken. Outlook-länkens parametrar ligger encodade i paramB64-parametern. I exemplet ovan är: aWRwPWh0dHBzOi8vd3d3LmV4YW1wbGUuY29tL2lkcA%3D%3D = idp=https://www.example.com/idp

Parametrar i paramB64 encodas enligt följande:

1. Base64 med padding
2. URLEncode

Använd förslagsvis Notepad++ med Base64-plugin eller liknande verktyg för encoding, och en icketekniker rekommenderas att kontakta sin supportleverantör för detta. När länken väl finns kan den självklart kopieras och användas hur många gånger som helst.

20.2. Normal eller embedded mode

Det finns två sätt att öppna nya fönster i en Office365-addin. Det ena är att använda vanligt Javascript (window.open), det andra är att använda Office egna funktioner (Office.context.ui.displayDialogAsync). TDialogs Office365-plugin har stöd för båda möjligheterna. Den genererade filen outlook-manifest (eller param-outlook-manifest) innehåller en sektion <!-- Normal mode --> som från början är bortkommenterad och en sektion <!-- Embedded mode --> som är aktiverad. Alltså är Embedded mode aktiverat per default och man kan aktivera Normal mode genom att aktivera den delen av XML:en och kommentera bort Embedded mode XML.

21. Provisionering av användare

TDialog är till största delen automatprovisionerande, dvs det använder attribut från IDP:n vid inloggningen för att automatiskt skapa upp en användaridentitet. Men, i fallet inloggning med E-legitimation finns ingen möjlighet att koppla personnummer med e-postadress och därför görs det normalt sett med ett inbjudningsmeddelande eller en självregistrering. Men, det finns också ett tredje sätt och det är att en administratör provisionerar användaren. Det går till så att administratören (med behörighet enligt 7.1 och med åtkomst enligt 2) laddar upp en fil i administratörsgränssnittet med informationen. Filen ska se ut enligt följande format:

191010101010;tian@test.se
191111111111;elvan@yahoo.se
191212121212;tolvan@gmail.com

22. SAML-inställningar

server.auth.internalEmailDomains=@grevlinge.se;@grevlinge2.se
Inställningen anger vilka mottagande e-postadresser som ska anses vara ”interna”. I exemplet ovan är alla e-postadresser som slutar på @grevlinge.se och @grevling2.se att betrakta som interna.
När meddelanden skickas till en användare av en intern e-postadress krävs inget personnummer eller liknande för att identifiera användaren, även om användaren ännu inte finns i TDialog. Istället fungerar det så att när användaren loggar in som krävs att IDP:n skickar med användarens e-postadress som ett attribut för att användaren ska får tillgång till meddelandet. Om någon skriver till magnus@grevlinge.se så kommer alltså den användare som loggar in och där IDP:n anger att användarens e-postadress är magnus@grevlinge.se att få se meddelandet. Dessutom krävs det att den IDP som verifierar användaren är en av IDP:erna som anges i inställningen server.auth.internalIdps.

server.auth.trustedEmailDomains=@myndigheten.se;@andramyndigheten.se
Inställningen anger vilka e-postadresser som ska anses vara ”trusted”. E-postadresser som är trusted är lite ett mellanting mellan interna och externa. En sådan användare betraktas som extern och har behörigheter som externa användare, och loggar in mot de externa snarare än interna IDP:erna. Men, användaren förväntas logga in mot en IDP som kan garantera användarens e-postadress, så det kommer inte upp någon uppmaning att fylla i personnummer om man skriver till en e-postadress som anses trusted.
Exempel: Låt säga att Grevlinge vill att anställda i myndigheten.se ska kunna logga in i lösningen, och myndigheten.se loggar in med EFOS-kort. På EFOS-korten finns en e-postadress, så dessa anställda behöver inte någon extra identifiering i form av personnummer eller liknande (och

loggar de in på ett sätt som inte ger någon e-postadress så kommer de inte komma in). Därför väljer Grevlinge kommun att sätta e-postadresser som slutar på @myndigheten.se som trusted.

server.auth.internalIdps=https://idp.grevlinge.se
Inloggning mot dessa IDP:er krävs för att få logga in som en intern användare. Syftet är att man inte ska kunna lura systemet att logga in som en intern användare med en extern IDP. Inställningen fylls i automatiskt vid installation.

server.saml.entityidguest= https://sp-guest.grevlinge.se
TDialog har två Service Providers: En för gästanvändare och en för interna användare. Inställningen anger Entity-ID för TDialogs Service Provider för gästanvändare. Inställningen sätts automatiskt vid installation.

server.saml.entityidfull= https://sp.grevlinge.se
TDialog har två Service Providers: En för gästanvändare och en för interna användare. Inställningen anger Entity-ID för TDialogs Service Provider för interna användare. Inställningen sätts automatiskt vid installation.

Att konfigurera signerade SAML-requests

Per default kör TDialog osignerade SAML-requests, dvs SAML-IDP producerar SAML-intyg utan att säkert ha verifierat att förfrågan är behörig. Säkerheten ligger istället i att den endast skickar över dessa intyg till en betrodd part, dvs en angripare kan inte utan ytterligare sårbarheter tillskansa sig ett intyg, och skulle det finnas sådana ytterligare sårbarheter i exempelvis användarens webbläsare är det oklart om det skulle skydda från att kvittera ut ett signerad förfrågan, då förfrågan till sin natur ändå skickas oautentiserat. I SAML 1 och om man använder oombedda intyg finns inget SAML-request alls, modellen är utvecklad för säkerhet oavsett om intyget är signerat eller inte.
Men om man så önskar kan TDialog köra signerade SAML-request, nedan ett exempel på hur det går till:

1) **Skapa självsignerat certifikat**
   Det går även att använda utgivna certifikat, men eftersom tilliten ligger mellan individuella certifikat snarare än mellan utgivare finns ingen större fördel med det.

   ```
   openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout
   mysitename.key -out mysitename.crt -config openssl.cnf
   ```

   (crt-filen kommer behövas i ett senare steg.)

   ```
   openssl pkcs12 -export -in mysitename.crt -inkey mysitename.key
   -out mysitename.p12 -name tdsamlsigningcert

   keytool -importkeystore -srckeystore mysitename.p12 -
   srcstoretype pkcs12 -srcalias tdsamlsigningcert -destkeystore
   mysitename.jks -deststoretype jks -destalias tdsamlsigningcert
   ```

Resultatet av kommandona är en JKS-fil med alias tdsamlsigningcert som innehåller en privat och publik nyckel, i detta exempel med lösenordet Vinter15.

2) **Ändra i securityContext.xml**

Ändra i keyManager. Om exempel-JKS:en skulle läggas i /tmp skulle konfigurationen se ut enligt nedan:

```
<bean id="keyManager"
class="org.springframework.security.saml.key.JKSKeyManager">
  <constructor-arg value="file:///tmp/mysitename.jks"/>
  <constructor-arg type="java.lang.String" value="Vinter15"/>
  <constructor-arg>
    <map>
      <entry key="tdsamlsigningcert" value="Vinter15"/>
    </map>
  </constructor-arg>
  <constructor-arg type="java.lang.String"
value="tdsamlsigningcert"/>
</bean>
```

Ändra org.springframework.security.saml.metadata.ExtendedMetadata för den SP man vill lägga till signing. Med exemplet ser konfigurationen ut enligt nedan. Notera att endast de relevanta parametrarna finns med i nedan lista:

```
<bean
class="org.springframework.security.saml.metadata.ExtendedMetad
ata">
  ...
  <property name="signingKey" value="tdsamlsigningcert"/>
  <property name="encryptionKey" value="tdsamlsigningcert"/>
  ...
</bean>
```

3) **Ändra metadatat**

Nu kommer SP:n signera SAML-requestet till IDP:n, så IDP:n behöver veta om det och behöver veta vilket med vilket certifikat den ska validera requestet. Enklast är att göra förändringen manuellt, men viktigt att det blir rätt. Lägg alltså in nedan XML i anslutning till SPSSODescriptor, öppna sedan mysitename.crt i en editor och kopiera hela innehållet till anvisad plats.

```
<md:SPSSODescriptor ...>
  <!--Tillägg-->
  <md:KeyDescriptor use="signing">
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:X509Data>
        <ds:X509Certificate><!-- Här lägger vi innehållet i
```

```
mysitename.crt --></ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </md:KeyDescriptor>
```

Sätt även AuthnRequestsSigned till true i metadatat.

4) Leverera det nya metadatat till IDP:n

23. Serverparametrar (host, port etc)

server.port=443

server.address=sakrameddelanden.grevlinge.se

Inställningarna ovan beskriver hur TDialog-servern ska bete sig från ett nätverksperspektiv. Vilket hostnamn ska den svara på, och på vilken port? I grundinställningarna finns inte server.address angiven, vilket gör att TDialog svarar på samtliga hostnamn, och defaultvärdet för server.port är 443.

24. Sök meddelanden

En användare med administratörsbehörighet kan söka efter meddelande-ID:n och därmed få upp meddelandets avsändare och mottagare. Detta är exempelvis användbart om en användare har fått ett notifieringsmeddelande med meddelandelänk men av olika anledningar inte kan logga in och läsa själva meddelandet. Då kan istället administratören se till vem eller vilka meddelandet var avsett och meddela dessa personer på annat sätt. Se 2 för åtkomst till administrationsgränssnittet och 7.1 för de behörigheter som krävs.

Nedan ett exempel på hur administratören kan ta reda på avsändare:

1) Någon har fått ett meddelande med meddelandelänk:
   https:/sakrameddelanden.kommun.se/messageId/3340/alias/guestUser
   Meddelande-ID:t är då 3340.
2) En administratör går till "Inställningar längst ner till höger i gränssnittet (finns tillgängligt om man är administratör).
3) Under "sök meddelande" matar administratören in meddelande-ID 3340 och trycker "sök".
4) Administratören får reda på meddelandets avsändare, mottagare och tidpunkt när meddelandet skickas.

25. Systemmeddelanden

server.maintenanceMessage kan användas för att skriva information som visas precis ovanför inkorgen/utkorgen/filerna för samtliga inloggade användare.

26. Säker digital kommunikation

För installation av accepunkt (Domibus) hänvisas till dokumentet ”TDialog - Installation av Domibus för Säker digital kommunikation”. Nedan beskrivs konfigurationen i TDialog för Säker digital kommunikation.

server.sdk.activated=true
Denna inställning styr om SDK-kopplingen alls är påslagen (default false).

server.sdk.httpsTrustStorePath= /trusteddialog/config/sdk-credentials/truststore.jks
Path till den truststore som innehåller certifikatet som gör att TDialog kan lita på webservicekopplingen mot sin Domibus. Domibusens https-certifikat ska alltså ligga här (se Domibus-dokumentationen).

server.sdk.cronParam=0/15 * * * * *
Intervall med vilket TDialogs SDK letar efter nya meddelanden i Domibus. Cron-format.

server.sdk.allowSendToUnknownFunctions=true
Inera tillhandahåller en adresslista i SDK-projektet, så att en organisation vet vilka kontaktvägar som finns i en annan organisation. Om denna inställning är true innebär det att en organisation får skicka meddelanden även till andra adresser än de som finns i adressboken (givet att även mottagande organisation godkänner, se server.sdk.allowReceiveToUnknownFunctions).

server.sdk.allowReceiveToUnknownFunctions=true
Inera tillhandahåller en adresslista i SDK-projektet, så att en organisation vet vilka kontaktvägar som finns i en annan organisation. Om denna inställning är true innebär det att TDialog godkänner att ta emot meddelanden även till andra adresser än de som finns i adressboken.

server.domibus.address=https://domibus.tdialog.com/domibus/services/backend
Adress till Domibusservern. Hostnamnet är det som rimligen ändras, pathen är densamma i Domibus

server.domibus.fromPartyId.value=domibus.tdialog.com
Det Domibus-ID som används på meddelanden från denna TDialog-installation. Måste matcha hostnamn och certifikatsalias för Domibusinstallationen (dokumentet ”TDialog - Installation av Domibus för Säker digital kommunikation”).

server.sdk.body.messageHeader.confidentiality=true
SDK-ramverket har stöd för att skicka en flagga för konfidentialitet med SDK-meddelanden. TDialog har i sig inte stöd för det, eftersom konfidentialitet är det normala i TDialog. Denna inställning anger om konfidentialitetsflaggan ska sättas på TDialog-meddelanden i SDK, inställningen blir alltså true eller false för samtliga meddelanden, det går i dagsläget inte att sätta på meddelandenivå.

server.sdk.body.organizationId.root=grevlinge.se
ID för organisationen. Måste matcha med Domibus och anmälan till Inera.

server.sdk.body.organizationId.label=Grevlinge kommun

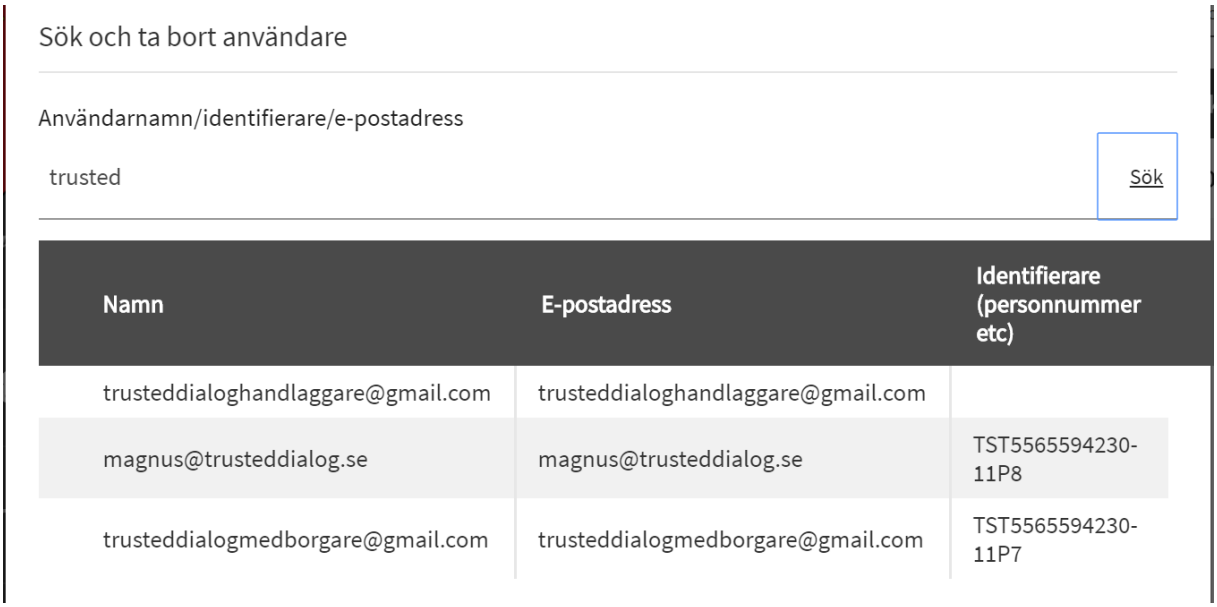
Namn på organisationen.

server.sdk.httpsKeyStorePath=/opt/tdialog/config/sdk-credentials/server.jks
Path till certifikat för Mutual TLS med Domibus. Kan vara samma som TDialogs servercertifikat.

Server.sdk.keyStorePassword=Vinter15
Lösenord för keystoret i server.sdk.httpsKeyStorePath.

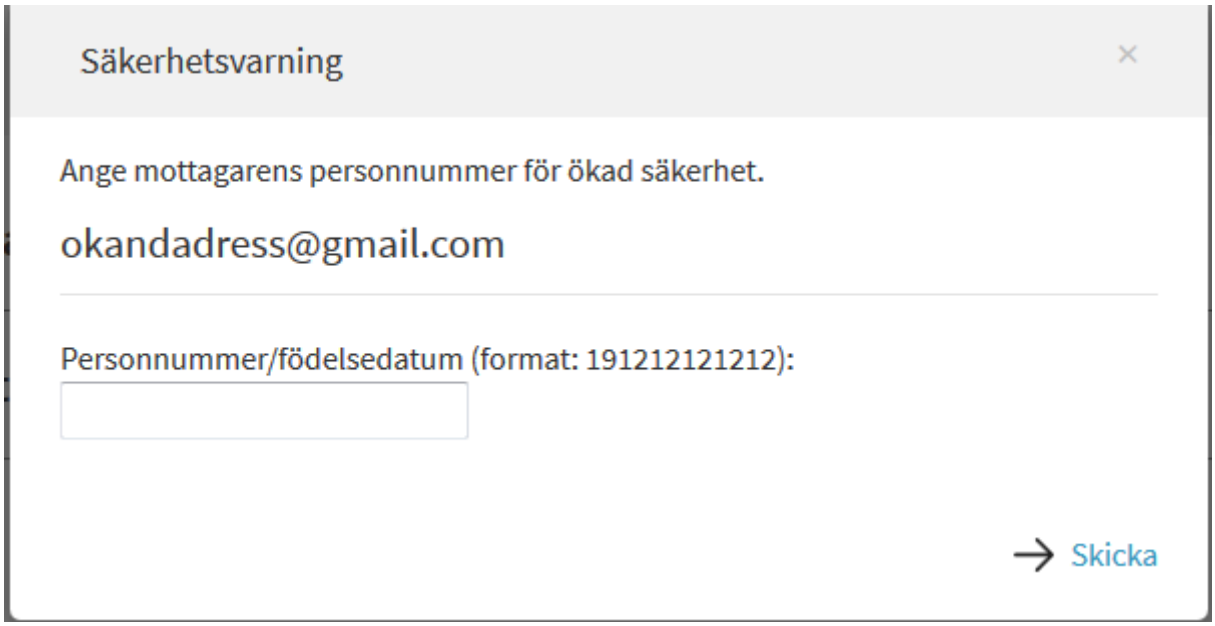
server.sdk.keyPassword=Vinter15
Lösenordet för nyckeln i server.sdk.httpsKeyStorePath.

27. Sök och radera användare



En användare med administratörsbehörighet kan söka efter och radera användare. Se 2 för åtkomst till administratörsgränssnittet och 7.1 för de behörigheter som krävs.
Man kan söka på namn, e-postadress eller ”identifierare”. Namnet kommer vara samma som e-postadressen för externa användare (som synes ovan) och identifierare kan vara personnummer, HSA-ID, sAMAccountName eller något annat beroende på vad som kommer in som attribut server.auth.userAttribute (se 4.1). Det går utmärkt att söka på en del av värdet och funktionen söker bland namn, e-postadresser och identifierare samtidigt.
Efter att användare sökts fram kan de även raderas. För att radera en användare, klicka på den och konfirmera att du vill ta bort användaren. Borttagande av användare är oåterkallerigt, och användarens samtliga filer och meddelanden tas då också bort. Observera att om någon skrivit ett meddelande till användaren så kommer den personens version av meddelandet att finnas kvar, precis som om det hade varit e-postmeddelanden.

28. Validering av mottagare



server.receiverValidation.validationRegex=^$|^[12]{1}[90]{1}[0-9]{2}[01]{1}[0-9]{1}[0-3]{1}[0-9]{1}|^[12]{1}[90]{1}[0-9]{2}[01]{1}[0-9]{1}[0-3]{1}[0-9]{1}[0-9]{4}$
Inställningen server.receiverValidation.validationRegex anger hur värdet i fältet ”personnummer/födelsedatum” ovan ska valideras. Observera att valideringen är ett Regular Expression (https://en.wikipedia.org/wiki/Regular_expression). Fördelen med det är att det går att göra mycket flexibla valideringsregler, nackdelen är att det kräver en viss matematisk/teknisk kunskap för att bemästra. Exemplet ovan betyder att fältet kräver att det antingen är tomt, innehåller ett födelsedatum på formatet 19121212 eller innehåller ett personnummer på formatet 191212121212.

server: receiverValidation: validationText: ”Personnummer skrivs på formatet 191212121212."
(config/application-prod.yml) Inställningen anger felmeddelandet om inmatningen i fältet inte är korrekt (enligt regeln server.receiverValidation.validationRegex ovan). Se 19.3 för hur inställningar i application-prod.yml fungerar.

29. Web Services

Från och med version 3.6 har TDialog stöd för REST web services. Det innebär att en extern part kan skicka TDialog-meddelanden genom att skicka (autentiserade) REST-meddelande. Med fullständig funktionalitet kommer i 3.7 och då även en mer omfattande dokumentation med exempelkod, men här visas den konfiguration som behöver göras i application-prod.properties för att aktivera REST web services.

server.integration.rest.enabled=true
Raden ovan krävs för att REST alls ska vara aktiverat i miljön. REST är per default avslaget.

server.integration.rest.trustStoreSettings=[{clientId: "testClient", publicKeyFile: "C:/td_workspace/trusteddialog/dev-only/config/api-credentials/public.der", permissions: "rwdi", sessionTime: “15”}]
trustStoreSettings är en JSON array av inställningar för varje klient som använder sig av REST. Varje klient har ett clientId, ett publikt certifikat (binärkodat DER) och en sträng med permissions. Permissions är enligt följande:
r, read: Klienten får läsa information ur TDialog
w, write: Klienten får skriva information till TDialog (dvs skicka meddelanden, skapa användare etc)
d, delete: Klienten får radera information (meddelanden, användare etc) från TDialog
i, impersonate: När klienten skriver något kan den göra det för någon annans räkning. En klient med impersonate kan exempelvis skicka ett meddelande från John till Lisa, medan en klient utan impersonate bara kan skicka ett meddelande från sig själv.

sessionTime betyder att klienten maximalt kan använda en JWT-token under så lång tid. Om sessionTime är det exempelvis en bra idé att använda tokens som med max 14 minuters giltighetstid, för att förhindra nekad åtkomst. Att använda sessioner ger något ökad prestanda på både klient och server (ca 20% enligt mätningar) men innebär även att om någon kommer över JWT-token under giltighetstiden kan den användas för olovliga operationer i TDialog.

server.integration.rest.tdAudience=td_test
Varje REST-anrop har en audience, och den måste matcha med audience som anges här för att anropet ska vara giltigt.

30. Whistleblower/utlämnande av handling

Från TDialog 3.9 finns möjlighet att skicka säkra meddelanden från anonyma användare genom att använda oinloggade formulär. Användarna kan även ta del av svar från handläggare genom att den anonyme användaren tilldelas ett lösenord som kan användas vid kommunikationen. Funktionen kallas Whistleblower, men den kan även användas för utlämnande av handling, förslagslåda, eller annat som kräver kommunikation med oinloggad användare.
OBS: TDialog ombesörjer att kommunikationen är anonym i applikationen. Det sätts inga kakor, loggas ingen information om klient osv i TDialog. För höga krav på anonymitet måste man dock se till att klientinformation inte lagras i andra delar av kundens serverinfrastruktur (lastbalansering, reverse proxy etc).
I whistleblower ingår exempelsidor med HTML och javascript, dessa finns i deploy/whistleblower. Om server.whistleblower.active sätts till true kommer dessa sidor vara tillgängliga som /whistleblower ute på Internet. Sidorna kan antingen lagras och ändras där eller läggas någon helt annan stans (på organisationens externwebb, exempelvis). De javascript som används för att anropa TDialog måste dock finnas intakta.
På serversidan konfigureras Whistleblower på följande sätt:

server.whistleblower.active=true
Aktiverar funktionen

server.whistleblower.htmlPath=/var/lib/trusteddialog/whistleblower/
Path till HTML:en. Denna kommer aktiveras om **server.whistleblower.active=true.**

server.whistleblower.cors.allowed-origins=www.grevlinge.se
Om HTML:en ligger på organisationens externwebb krävs att externwebbens adress anges här. Detta gör att användarens webbläsare tillåter nedladdning via Javascript från externwebbens adress till TDialog.

server.whistleblower.allowedRecipients=[{"name": "whistleblower@grevlinge.se", "singleMessageOnly": "true", "allowReply": "true", "mailNotification": "false", "recipients": [{"address": "socialtjanst@grevlinge.se"},{"address": "kommunledning@grevlinge.se"}]}, {"name": "handlingar@grevlinge.se ", "mailNotification": "true", "allowReply": "true", "singleMessageOnly": "false", "recipients": [{"address": "registrator@grevlinge.se"}]}]
Exemplet ovan visar hur vi konfigurerat ett formulär med adress whistleblower@grevlinge.se som kan skicka till socialtjanst@grevlinge.se och kommunledning@grevlinge.se, och ett formulär med adress handlingar@grevlinge.se som kan skicka meddelanden till registrator@grevlinge.se. Nedan beskrivs varje del av konfigurationen:
allowedRecipient består av en array av formulär, dvs i exemplet är det ett formulär som används för visselblåsning och ett som används för utlämnande av handling. Varje formulär har sedan ett antal parametrar:
name: Formulärets namn (en e-postadress)
singleMessageOnly: Om true ser den anonyme användaren bara senaste svaret som skickats, annars hela konversationen.
allowReply: Om true kan whistleblower-meddelanden besvaras
mailNotification: Om true skickas notifieringar från whistleblower-meddelanden.
recipients: Lista med adresser till tillåtna mottagare.