Generalitat de Catalunya
Centre de Telecomunicacions
i Tecnologies de la Informació

Framework Corporatiu J2EE

Servei de correu electrònic

Versió 1.2

Barcelona, 13 / febrer / 2007

Històric de modificacions

Data	Autor	Comentaris	Versió
09/12/2005	Atos Origin, sae openTrends	Versió inicial del document	1.0
03/01/2006	Atos Origin, SAE	Versió 1.1 d'OpenFrame	1.1
13/02/2007	Atos Origin, SAE	Versió 1.2 d'OpenFrame	1.2

Llegenda de Marcadors

Índex

1. Introducció

1.1. Propósit

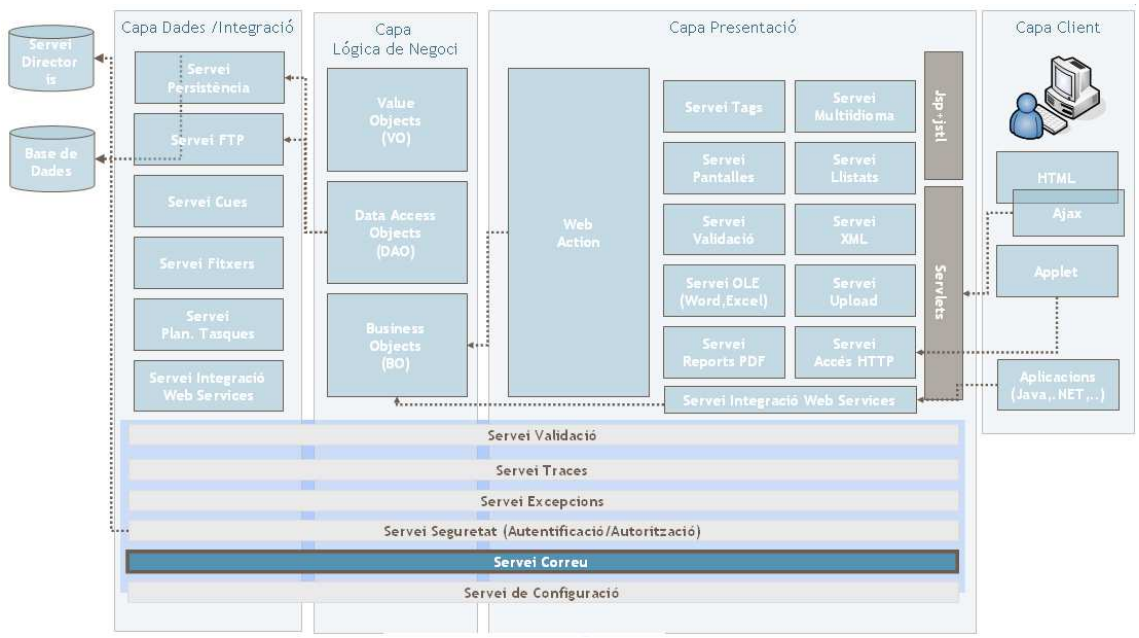
Aquest servei té com a objectiu permetre l'enviament de correus electrònics a una o diverses adreces especificades a qualsevol dels següents recipients:

- Destinataris principals
- Destinataris secundaris
- Destinataris ocults

Permet diferents modes d'enviament, tant en text pla, com en mode HTML, i en tots 2 casos oferint la possibilitat d'adjuntar un o més fitxers.

1.2. Context i Escenaris d'Ús

El Servei de Correu es troba dins dels serveis de Propósit General de openFrame.



1.3. Versions i Dependències

En el present apartat es mostren quines són les versions i dependències necessàries per fer ús del Servei.

1.3.1. Versions

No s'han produït canvis respecte la versió 1.1.

1.3.2. Dependències Bàsiques

Nom	Tipus	Versió	Descripció
activation	jar	1.0.2	
mail	jar	1.3.3	
openFrame-services-configuration	jar	1.0-SNAPSHOT	
openFrame-services-exceptions	jar	1.0-SNAPSHOT	
openFrame-services-logging	jar	1.0-SNAPSHOT	
spring	jar	1.2.5	http://www.springframework.org

També cal considerar les dependències associades a l'ús dels serveis de openFrame (consultar els documents associats).

1.3.3. Dependències Addicionals

- Proves Unitàries del Servei

Nom	Tipus	Versió	Descripció
junit	jar	3.8.1	

1.4. A qui va dirigit

Aquest document va dirigit als següents perfils:

- Programador. Per conèixer l'ús del servei
- Arquitecte. Per conèixer quins són els components i la configuració del servei
- Administrador. Per conèixer com configurar el servei en cadascun dels entorns en cas de necessitat

1.5. Documents i Fonts de Referència

[1] Spring Mail http://www.springframework.org/docs/reference/mail.html

1.6. Glossari

2. Descripció Detallada

2.1. Arquitectura i Components

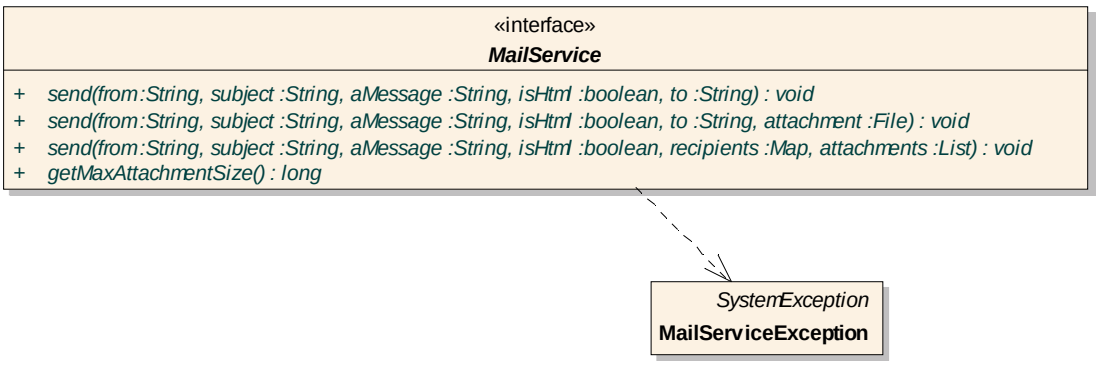
openFrame ofereix la possibilitat d'utilitzar diferents implementacions del Servei de Correu. Dins la filosofia general d'oferir interfícies, els clients no es veurien afectats per un canvi d'implementació.

Els components podem classificar-los en:

- Interfícies i Components Genérics. Interfícies del servei i components d'ús general amb independència de la implementació escollida.
- Implementació de les interfícies basada en Spring i JavaMail

2.1.1. Interfícies i Components Genérics

El Servei de Correu defineix les següents interfícies:



Component	Package	Descripció
MailService	net.opentrends.openframe.services.mail	Interfície del Servei de Correu
MailServiceException	net.opentrends.openframe.services.mail.exception	Excepció del Servei de Correu

2.1.2. Components Implementació Spring amb JavaMail

Component	Package	Descripció
SpringMailServiceImpl	net.opentrends.openframe.services.mail.impl	Implementació del Servei de Correu basada en Spring i JavaMail

Component	Package	Descripció
SpringJavaMailSender Impl	net.opentrends.openframe.services .mail.impl	Delegat de la implementació del Servei que permet l'enviament amb diferents tipus de missatges (Spring i JavaMail)
SmartMimeMessage	net.opentrends.openframe.services .mail.impl	Subclasse de la classe 'MimeMessage' de JavaMail

Veure javadoc del Servei per a més referéncia.

2.2. Instal·lació i Configuració

2.2.1. Instal·lació

La instal·lació del servei requereix de la utilització de la llibreria 'openFrame-services-mailing' i les dependències indicades a l'apartat 'Introducció-Versions i Dependències'.

2.2.2. Configuració

La configuració del Servei de Correu implica els següents pasos:

1) Definir el delegador d'enviaments
2) Definir el servei
3) Definir les propietats del servei
4) Definir els literals multiidioma de les excepcions

Definició del Delegador d'Enviaments

```
<bean id="mailSender"
      ...>
```

Fitxer de configuració: openFrame-services-mailing.xml

Ubicació proposada: <PROJECT_ROOT>/src/main/resources/spring

Es permet l'ús de la classe 'org.springframework.mail.javamail.JavaMailSenderImpl'. Aquesta defineix les següents propietats:

Propietats:

Propietat	Requerit	Descripció
host	Sí	Nom del servidor de correu sortint (smtp)
port	No	Port del servidor de correu sortint (smtp)
username	No	Usuari de connexió al servidor de correu sorting (smtp)
password	No	Password de l'usuari de connexió

Exemple:

```
<bean id="mailSender"
class="net.opentrends.openframe.services.mail.impl.SpringJavaMailSenderImpl">
          <property name="host" value="${mailSender.host}"/>
          <property name="port" value="${mailSender.port}"/>
          <property name="username" value="${mailSender.username}"/>
          <property name="password" value="${mailSender.password}"/>
</bean>
```

Definició del Servei

```
<bean id="mailService"
      ...>
```

Fitxer de configuració: openFrame-services-mailing.xml

Ubicació proposada: <PROJECT_ROOT>/src/main/resources/spring

En l'actualitat s'ofereix la implementació 'net.opentrends.openframe.services.mail.impl.SpringMailServiceImpl', en la que definirem les següents propietats:

Propietat	Requerit	Descripció
mailSender	Sí	Referència a l'enviador
logService	No	Referència al Servei de Traces
maxAttachmentSize	No	Tamany màxim permés dels fitxers adjunts

Exemple:

```
<bean id="mailService"
class="net.opentrends.openframe.services.mail.impl.SpringMailServiceImpl">
   <property name="maxAttachmentSize" value="${mailService.maxAttachmentSize}"/>
   <property name="mailSender"><ref bean="mailSender"/></property>
   <property name="logService"><ref bean="loggingService"/></property>
</bean>
```

Definició de les Propietats del Servei

```
mailSender.host=...
...
```

Fitxer de configuració: mail.properties

Ubicació proposada: <PROJECT_ROOT>/src/main/resources/mail

En aquest fitxer definirem les propietats del servei. Aquestes propietats corresponen amb les que hem definit a les configuracions de l'enviador i del servei.

Com a recordatori, tal i com s'ha vist al 'Servei de Configuració' podem fer ús de la classe 'HostPropertyPlaceHolderConfigurer' per definir tots els nostres fitxers de propietats.

En el cas del Servei de Correu podem seguir la mateix filosofia, tal i com es mostra en el següent exemple:

```
<bean id="configurationService"
class="net.opentrends.openframe.services.configuration.springframework.beans.factory.config.HostPropertyPlaceholderConfigurer">
  <property name="basePropertyFiles">
   <list>
          <value>classpath:jdbc/jdbc.properties</value>
          <value>classpath:mail/mail.properties❶</value>
          <value>classpath:file/fileUploadService.properties</value>
          <value>classpath:file/fileService.properties</value>
   </list>
  </property>
</bean>
```

❶ Referència al fitxer de propietats 'mail.properties', que estarà ubicat al directori '/mail' dins el classpath

En l'exemple el configurador fa ús d'un localitzador de fitxers depenent del host (*HostPropertyResourceConfigurer*). Això vol dir que si en aquest configurador incloem una referència a un fitxer de propietats anomenat " mail.properties", realment es buscarà un fitxer anomenat "mail.properties.nom_del_host". No obstant, aquesta no és l'única possibilitat, ja que es pot fer servir un altre configurador que no depengui del host (com per exemple *PropertyOverrideConfigurer*). Per a més informació sobre els configuradors veure el document del **Servei de Configuració.**

Exemple de fitxer de propietats:

```
mailSender.host=smtp.es.int.atosorigin.com
mailSender.port=25
mailSender.username=manel.mateos
mailSender.password=
mailService.maxAttachmentSize=1024
```

Definició dels Missatges d'Error del Servei

```
openFrame.services.mail.attachment_size_exceeded=...
...
```

Fitxer de configuració: openFrame-errorCodes.properties

Ubicació proposada: <PROJECT_ROOT>/src/main/resources/i18n/errorCodes.properties

Configurar els següents codis d'error:

```
openFrame.services.mail.attachment_size_exceeded=Attachment size exceeded: {0}
bytes (max: {1})
openFrame.services.mail.error_preparing_addresses=Error preparing addresses
(to,bcc,cc)
openFrame.services.mail.error_sending_mail=Error sending mail
```

2.3. Utilització del Servei

2.3.1. Enviament de Correus

Per enviar correus, les classes client definiran un atribut 'mailService' de tipus 'MailService'. Mitjançant el Servei de Configuració s'especificarà quina és la implementació escollida (transparent al client).

Es permeten diferents combinacions d'enviament de correus (per a més referència consultar el javadoc de la interfície 'MailService). A mode de resum, podem enumerar:

- `public void send(String from, String subject, String aMessage, boolean isHtml, String to)`

  Envia un missatge ("*aMessage*") des d'una direcció a una altra ("*from*","*to*") amb un tópic ("*subject*") en format de text pla o HTML ("*isHtml*").
- `public void send(String from, String subject, String aMessage, boolean isHtml, String to, File attachment)`

  Envia un missatge ("*aMessage*") des d'una direcció a una altra ("*from*","*to*") amb un tópic ("*subject*") en format text pla o HTML ("*isHtml*") amb un fitxer annex ("*attachment*")

- `public void send(String from, String subject, String aMessage, boolean isHtml, String to, Map recipients, List attachment)`

  Envia un missatge ("*aMessage*") des d'una direcció a unes altres ("*from*","*recipients*") amb un tópic ("*subject*") en format text pla o HTML ("*isHtml*") amb una llista (o sense llista) de fitxers annexes ("*attachments*"). La llista de direccions annexes té forma de taula ("*java.util.Map*") on els continguts d'aquesta taula són:

  - Clau: pot ser un *javax.mail.Message.RecipientType.TO*, un *javax.mail.Message.RecipientType.BCC* ó *javax.mail.Message.RecipientType.CC*
  - Valor: és un literal ("*String*") o una llista de literals ("*String[]*")

  La llista d'anexes és una llista de fitxers ("*java.io.File*"). Si és nul·la no s'enviarà cap fitxer anexe.

2.4. Eines de Suport

2.5. Integració amb Altres Serveis

2.6. Preguntes Freqüènts

3. Exemples

3.1. Exemple de Prova Unitària

Un exemple d'utilització del servei de correu són els tests unitaris, a on s'obté el bean del servei a partir del fitxer de definició (applicationContext.xml) i s'envia un correu electrònic a les adreces de test especificades.

```
package net.opentrends.openframe.services.mail.test;

...

public class MailServiceTest extends TestCase {
...
    public void testMailingWithoutAttachment(){
/**
 * Obtenim les definicions dels beans utilitzats
 */
        BeanFactory beanFactory = new
ClassPathXmlApplicationContext("applicationContext.xml");
/**
 * Obtenim el servei de correu
 */
        MailService mailService = (MailService)
beanFactory.getBean("mailService");
/**
 * Obtenim les adreces de test
 */
        TestClient clientTest = (TestClient) beanFactory.getBean("clientTest");
/**
 * Obtenim el servei de logs per deixar traces
 */
        LoggingService logService = (LoggingService)
beanFactory.getBean("logService");


        try {
           logService.getLog(this.getClass()).debug("From
test="+clientTest.getFromTest()+",To test="+clientTest.getToTest());
/**
 * Enviem el correu de forma fàcil i senzilla
 */
           mailService.send(clientTest.getFromTest(),"MailServiceTest "+new
Date(),"MailServiceTest without attachment",false,clientTest.getToTest());
        }
        catch (Exception e) {...}
    }
...
}
```

4. Annexos