

## **MailProvider (Beispiele)**

# Verschicken einer Email

Methode	Beispiel
send	<pre>package org.nuclet.businessentity;  import org.nuclet.printout.FormularAbschlussAuftragPO; import org.nuclos.api.common.NuclosFile; import org.nuclos.api.mail.NuclosMail; import org.nuclos.api.provider.PrintoutProvider; import org.nuclos.api.provider.MailProvider;  public class AbschlussAuftragRegel implements InsertFinalRule {      public void insertFinal(InsertContext context) throws BusinessException {          Auftrag a = context.getBusinessObject(Auftrag.class);          NuclosFile result = PrintoutProvider.run(FormularAbschlussAuftragPO.Deutsch_PDF, a. getId());          NuclosMail newMail = new NuclosMail();          newMail.setSubject("Abschluss eines Auftrags");         newMail.setFrom("automatic@meineFirma.de");         newMail.addRecipient("Max.Mustermann@meinKunde.de");         newMail.setMessage("Der Auftrag wurde erfolgreich abgeschlossen.");         newMail.addAttachment(result);          MailProvider.send(newMail);     } }</pre> <p>Erläuterung:</p> <p>Mit dem erfolgreichen Abschluss eines Auftrags soll automatisch eine Bestätigungsmail an den Kunden verschickt werden. Das vom PrintoutProvider erzeugte Formular wird dabei als NuclosFile an die Email gehängt.</p> <p>Anzumerken ist, dass durchaus mehrere NuclosFiles hinzugefügt werden können. Auch die Angabe von BCC- und CC-Empfängern ist möglich.</p> <p>Im Falle eines Laufzeit-Fehlers wird eine BusinessException geworfen.</p>

receive

```
package org.nuclet.businessentity;

import java.util.List;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.common.NuclosFile;
import org.nuclos.api.context.GenerateContext;
import org.nuclos.api.exception.BusinessException;
import org.nuclos.api.mail.NuclosMail;
import org.nuclos.api.provider.MailProvider;
import org.nuclos.api.rule.GenerateRule;
/** @name
 * @description
 * @usage
 * @change
 */
@Rule(name="GetEmails", description="GetEmails")
public class GetEmails implements GenerateRule {
    public void generate(GenerateContext context) throws BusinessException {
        Materialbestellung bestellung = context.getTargetObject(Materialbestellung.class);
        List<NuclosMail> allMails = MailProvider.receive(true);

        for (NuclosMail mail : allMails) {
            for (NuclosFile file : mail.getAttachments()) {
                bestellung.insertAttachment(
                    file, "Bestätigungsmail von '" + mail.getFrom() +
                    "' mit Betreff '" + mail.getSubject() + "'");
            }
        }
    }
}
```

Damit Emails von einem Server abgeholt werden können, müssen die relevanten SMTP - Daten in den Systemparametern innerhalb Nuclos hinterlegt werden. (siehe unten aufgeführten Link).

Im oberen Beispiel wird im Methodenaufruf *MailProvider.receive(true)* der Parameter "true" mitgeliefert, was bedeutet, dass die Emails nach dem Abholen auf dem Server gelöscht werden. Sollte dies nicht erwünscht sein, können mit der Angabe von *false* alle Emails erhalten bleiben, werden aber beim nächsten Aufruf von *receive()* wieder mitgeliefert!

Im weiteren Verlauf werden alle gefundenen Emails durchlaufen und Anhänge, falls vorhanden, an das neu generierte Objekt *Materialbestellung* angehängt (Dokumentenanhäng). Dabei werden Mail-relevante Informationen in den Kommentarblock eingefügt.

Systemparameter zur Konfiguration des Emailservers siehe [Parameter](#)