

MailProvider

🌐 Sprache: **Deutsch** · [English](#)

MailProvider

E-Mails aus Regeln senden und empfangen – `send()`, `receive()` und die `NuclosMail`-Klasse.

REFERENZ

ENTWICKLER

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

Auf dieser Seite

- [Wozu dient der MailProvider?](#)
- [send\(mail\)](#)
- [receive\(bDeleteMails\)](#)
- [NuclosMail](#)
- [Verwandte Seiten](#)

Wozu dient der MailProvider?

Der MailProvider stellt Methoden zum **Versenden und Abholen** von E-Mails bereit.

send(mail)

Verschickt eine `NuclosMail`.

```
public static void send(NuclosMail mail) throws BusinessException;
```

```
package org.nuclet.businessentity;

import org.nuclet.printout.FormularAbschlussAuftragP0;
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(FormularAbschlussAuftragP0.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);
    }
}
```

receive(bDeleteMails)

Holt E-Mails vom Server. Mit `true` werden sie nach dem Abruf gelöscht, mit `false` bleiben sie erhalten (und kommen beim nächsten Aufruf erneut).

```
public static List<NuclosMail> receive(boolean bDeleteMails) throws BusinessException;
```

```
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() + "'");
            }
        }
    }
}
```

NuclosMail

Die API-Klasse NuclosMail nimmt alle Bestandteile einer Standard-E-Mail auf:

Attribut	Bedeutung
From	Absender (Default aus Systemparameter <i>SMTP Sender</i>).
Recipients / CC / BCC	Empfänger als String, mehrere per Semikolon getrennt.
Subject / Message	Betreff und Inhalt.
ReplyTo	Antwortadresse.
Headers	z.B. Charset/Content-Type (Default text/plain; charset=UTF-8).
Attachment	Liste von NuclosFile-Anhängen.



Serverkonfiguration

SMTP/POP3-Zugangsdaten werden in den Systemparametern hinterlegt (Administration Parameter).

Verwandte Seiten



QueryProvider

Daten laden.

[Öffnen](#)



BusinessObjectProvider

BOs schreiben.

[Öffnen](#)



DatasourceProvider

Datenquellen ausführen.

[Öffnen](#)

