



Softwarepraktikum im Grundstudium im Wintersemester 2002/2003

8. Übung, 23. Januar 2003

1. Aufgabe (Verschicken von E-Mails)

Implementieren Sie, dass wenn eine Aufgabe angelegt oder eine bestehende Aufgabe (Kommentar, Benutzer, Status, etc.) verändert wird, der zugeordnete Bearbeiter per Email benachrichtigt wird. Bei der Zuordnung eines neuen Bearbeiters zu einer bereits existierenden Aufgabe sollen sowohl der alte als auch der neue Bearbeiter benachrichtigt werden. Alle Mails sollen auch an alle Manager versandt werden.

Zum Verschicken der E-Mails verwenden wir das JavaMail API (<http://java.sun.com/products/javamail/>). Bitte lesen Sie den Developer Connection Tech Tip zu diesem Thema unter:

<http://developer.java.sun.com/developer/JDCTechTips/2001/tt1023.html>.

- Schreiben Sie eine Klasse MailSender zum Verschicken von Benachrichtigungen. Verwenden Sie einen Konstruktor in der unten angegebenen Art. Der Konstruktor bekommt einen Iterator mit den User-Objekten für die Empfänger übergeben.
- Ersetzen Sie den smtp-host durch einen von Ihnen verwendbaren und stellen Sie als From-Adresse Ihre eigene Mailadresse ein.

```
public MailSender(Iterator recipients) {
    try {
        props = new Properties();
        props.put("mail.smtp.host", "mail.gmx.de");//SMTP-Host ändern!
        props.put("mail.smtp.auth", "true");//Authentication?
        session = Session.getDefaultInstance(props, new MyAuthenticator());
        message = new MimeMessage(session);

        message.setFrom(...);//Mail-Adresse Eintragen

        while (recipients.hasNext()) {
            User u = (User) recipients.next();
            message.addRecipient(...);
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- Sie benötigen die Zeile “props.put(“mail.smtp.auth”....) nur dann, wenn Sie einen SmtP-Server mit Autorisierung verwenden. Implementieren Sie die Klasse MyAuthenticator als Subklasse von Authenticator, um Benutzername und Passwort an den SmtP-Server zu verschicken.
- Überlegen Sie sich eine geeignete Datenstruktur, in der Sie die Änderungen an einer Aufgabe (Kommentare nicht vergessen!) verwalten. Diese Datenstruktur soll bei den verändernden Methoden im StorageAccess gefüllt und als Rückgabewert zurückgegeben werden. Tipp: Die Datenstruktur sollte eine Menge von Tupeln aus

(Feldname, alter Wert, neuer Wert) enthalten. Bei neu angelegten Aufgaben bzw. Kommentaren sind die Einträge für den alten Wert null.

- Implementieren Sie außerdem eine Methode `sendMail`, welche die Mail an alle Empfänger verschickt. Der Mailtext sollte die Änderungen übersichtlich darstellen.
- Erweitern Sie die entsprechenden Servlets so, dass Benachrichtigungen versandt werden.

Wichtiger Hinweis: Verwenden Sie zum Testen bitte nur tatsächlich existierende Mailadressen, weil Sie sonst Fehler vom SmtP-Server bekommen. Versuchen Sie außerdem beim Testen möglichst wenige Mitmenschen zu „nerven“ ☺. Sollte das Verschicken der Mails nicht gleich funktionieren, bauen Sie bitte die folgende Anweisung in den MailSender-Konstruktor ein: `session.setDebug(true)`; Dadurch erhalten Sie Zugriff auf die interne Kommunikation zwischen Ihrer Applikation und dem verwendeten SmtP-Server. Diese wird in der Logdatei `Catalina.out` aufgezeichnet.

Testat

Donnerstag, 6.02.2003 von 10.00 - 11.45 Uhr