

E-Mail-Konfiguration - planbar

Übersicht

planbar verwendet **drei verschiedene E-Mail-Adressen** für unterschiedliche Zwecke:

E-Mail-Adresse	Zweck	SMTP-Account
noreply@planbar.dev	Ticket/Subtask-Benachrichtigungen	SMTP_HOST
noreply@planbar.dev	Login-Benachrichtigungen	SMTP_LOGIN_HOST
update@planbar.dev	Tägliche/Wöchentliche Updates	SMTP_UPDATE_HOST

SMTP-Konfiguration (.env)

Vollständige Konfiguration

```
# === Ticket/Subtask-Benachrichtigungen (noreply@planbar.dev) ===
SMTP_HOST=asmtplib.mail.hostpoint.ch
SMTP_PORT=587
SMTP_USER=noreply@planbar.dev
SMTP_PASSWORD=6ub!k!X.g*!FYh!
SMTP_FROM=noreply@planbar.dev

# === Login-Benachrichtigungen (noreply@planbar.dev) ===
SMTP_LOGIN_HOST=asmtplib.mail.hostpoint.ch
SMTP_LOGIN_PORT=587
SMTP_LOGIN_USER=noreply@planbar.dev
SMTP_LOGIN_PASSWORD=6ub!k!X.g*!FYh!
SMTP_LOGIN_FROM=noreply@planbar.dev

# === Update-E-Mails (update@planbar.dev) ===
SMTP_UPDATE_HOST=asmtplib.mail.hostpoint.ch
SMTP_UPDATE_PORT=587
SMTP_UPDATE_USER=update@planbar.dev
SMTP_UPDATE_PASSWORD=i@8RAU9_v2Qoyu3
SMTP_UPDATE_FROM=update@planbar.dev

# === Firmenname und Design ===
COMPANY_NAME=planbar
PRIMARY_COLOR=#3b82f6
NEXTAUTH_URL=https://planbar.dev
```

Sicherheitshinweis

WICHTIG: Diese Credentials **NIEMALS** in Git committen!

- .env ist bereits in .gitignore

- Auf Produktions-Servern: Umgebungsvariablen über Server-Konfiguration setzen
 - Bei Vercel/Netlify: Environment Variables in Settings
-

E-Mail-Typen

1. Ticket/Subtask-Benachrichtigungen (noreply@planbar.dev)

Wann werden sie versendet?

- Neues Ticket erstellt und zugewiesen
- Ticket wird neu zugewiesen
- Ticket-Status ändert sich
- Neuer Subtask wird zugewiesen
- Subtask wird neu zugewiesen
- Subtask wird als erledigt markiert

Code:

```
import { sendTicketAssignedEmail, sendSubTaskAssignedEmail } from '@/lib/email';

// Beispiel: Ticket zugewiesen
await sendTicketAssignedEmail(
  assigneeEmail,
  assigneeName,
  ticketTitle,
  ticketId,
  assignedBy
);
```

Benutzer-Steuerung:

- Benutzer können diese in ihrem **Profil** aktivieren/deaktivieren
 - Feld: `emailNotifications` (Boolean)
 - Standard: `true`
-

2. Login-Benachrichtigungen (noreply@planbar.dev)

Wann werden sie versendet?

- Bei jedem erfolgreichen Login
- Automatisch im NextAuth JWT-Callback

Inhalt:

-  Bestätigung der Anmeldung
-  Zeitpunkt der Anmeldung
-  IP-Adresse (falls verfügbar)
-  Geräteinfo (falls verfügbar)
-  Sicherheitshinweis

Code:

```
// Automatisch in lib/auth.ts (JWT Callback)
import { sendLoginNotificationEmail } from '@/lib/email';

await sendLoginNotificationEmail(
  user.email,
  user.name,
  new Date(),
  ipAddress,
  userAgent
);
```

Benutzer-Steuerung:

- **Kann nicht deaktiviert werden** (Sicherheitsfunktion)
 - Wird bei jedem Login automatisch versendet
-

3. Tägliche/Wöchentliche Updates (update@planbar.dev)

Wann werden sie versendet?

- **Täglich:** Jeden Tag zur gleichen Zeit
- **Wöchentlich:** Jeden Montag zur gleichen Zeit
- Nur an Benutzer, die es aktiviert haben

Inhalt:

- Statistiken: Neue Projekte, neue Tasks, erledigte Tasks
- Bald fällige Tasks (nächste 7 Tage)
- Neue Zuweisungen (Tickets + Subtasks)

Manueller Versand (Test):

```
# Tägliches Update testen
curl http://localhost:3000/api/send-update-emails?frequency=daily&test=true

# Wöchentliches Update testen
curl http://localhost:3000/api/send-update-emails?frequency=weekly&test=true
```

Benutzer-Steuerung:

- Benutzer können Häufigkeit im **Profil** einstellen
 - Feld: `emailReportFrequency` (String)
 - Optionen: `"none"`, `"daily"`, `"weekly"`
 - Standard: `"none"`
-



Automatisierung (Cron-Jobs)

Vercel Cron-Jobs

Datei: `vercel.json`

```
{
  "crons": [
    {
      "path": "/api/send-update-emails?frequency=daily",
      "schedule": "0 8 * * *"
    },
    {
      "path": "/api/send-update-emails?frequency=weekly",
      "schedule": "0 8 * * 1"
    }
  ]
}
```

Erklärung:

- `0 8 * * *` = Täglich um 8:00 Uhr UTC
- `0 8 * * 1` = Jeden Montag um 8:00 Uhr UTC

Hinweis: Vercel Cron-Jobs sind nur in Pro/Team-Plänen verfügbar!

Alternative: GitHub Actions

Datei: `.github/workflows/send-emails.yml`

```

name: Send Update Emails

on:
  schedule:
    - cron: '0 8 * * *' # Täglich 8:00 UTC
    - cron: '0 8 * * 1' # Montags 8:00 UTC

jobs:
  send-emails:
    runs-on: ubuntu-latest
    steps:
      - name: Send Daily Updates
        run: |
          curl -X GET "https://planbar.dev/api/send-update-emails?frequency=daily"

      - name: Send Weekly Updates (Mondays only)
        if: github.event.schedule == '0 8 * * 1'
        run: |
          curl -X GET "https://planbar.dev/api/send-update-emails?frequency=weekly"
```

Alternative: Externe Cron-Dienste

Empfohlene Dienste:

- cron-job.org (<https://cron-job.org>) (kostenlos)
- [EasyCron](https://www.easycron.com) (<https://www.easycron.com>) (kostenlos bis 250 Jobs/Monat)
- [Cronitor](https://cronitor.io) (<https://cronitor.io>) (kostenpflichtig)

Setup:

1. Account erstellen
2. Cron-Job anlegen:
 - URL: <https://planbar.dev/api/send-update-emails?frequency=daily>
 - Schedule: `0 8 * * *` (täglich 8:00 Uhr)
3. Zweiten Job anlegen:

- URL: `https://planbar.dev/api/send-update-emails?frequency=weekly`
 - Schedule: `0 8 * * 1` (montags 8:00 Uhr)
-

Testen

1. Test-E-Mail (SMTP-Konfiguration prüfen)

```
# Test-E-Mail an eigene Adresse senden
curl "http://localhost:3000/api/test-email?to=ihre-email@example.com"
```

Ergebnis:

```
{
  "success": true,
  "message": "Test-E-Mail erfolgreich versendet!",
  "recipient": "ihre-email@example.com",
  "smtpConfig": {
    "host": "asmtpt.mail.hostpoint.ch",
    "port": "587",
    "user": "noreply@planbar.dev",
    "from": "noreply@planbar.dev"
  }
}
```

2. Login-Benachrichtigung testen

1. Öffnen Sie die App: `http://localhost:3000`
2. Melden Sie sich ab (falls angemeldet)
3. Melden Sie sich neu an
4. E-Mail sollte an Ihre Adresse gesendet werden

Prüfen Sie:

-  Postfach: E-Mail von `noreply@planbar.dev`
-  Zeitpunkt stimmt
-  Link zum Profil funktioniert

3. Update-E-Mail testen (manuell)

```
# Tägliches Update (Test-Modus)
curl "http://localhost:3000/api/send-update-emails?frequency=daily&test=true"

# Wöchentliches Update (Test-Modus)
curl "http://localhost:3000/api/send-update-emails?frequency=weekly&test=true"
```

Voraussetzung:

- Sie müssen eingeloggt sein
- In Ihrem Profil muss `emailReportFrequency` auf `"daily"` oder `"weekly"` gesetzt sein

Ergebnis:

```
{
  "success": true,
  "testMode": true,
  "user": "ihre-email@example.com",
  "userId": "...",
  "email": "ihre-email@example.com",
  "sent": true,
  "stats": {
    "newTickets": 2,
    "newSubTasks": 5,
    "completedSubTasks": 3,
    "dueSoonSubTasks": 2
  }
}
```

4. Ticket/Subtask-Benachrichtigung testen

1. Erstellen Sie ein neues Ticket
2. Weisen Sie es einem anderen Benutzer zu
3. Der Benutzer sollte eine E-Mail erhalten

Oder:

1. Öffnen Sie ein Ticket
2. Fügen Sie einen Subtask hinzu
3. Weisen Sie den Subtask einem Benutzer zu
4. Der Benutzer sollte eine E-Mail erhalten



Fehlerbehebung

⚠ “Email transporter (login) not configured”

Problem: SMTP_LOGIN_* Variablen fehlen oder sind falsch.

Lösung:

1. Überprüfen Sie `.env` :

```
bash
grep SMTP_LOGIN /home/ubuntu/github_repos/planbar/nextjs_space/.env
```

2. Stellen Sie sicher, dass alle Variablen gesetzt sind
3. Server neu starten

⚠ “Authentication failed” (SMTP)

Problem: Falsches Passwort oder Server.

Lösung:

1. Prüfen Sie die Credentials:
 - `SMTP_USER` : `noreply@planbar.dev`
 - `SMTP_PASSWORD` : `6ub!k!X.g*!FYh!`

- SMTP_HOST : assmtp.mail.hostpoint.ch
- SMTP_PORT : 587

1. Testen Sie manuell mit `telnet` :

```
bash
telnet assmtp.mail.hostpoint.ch 587
```

Keine Login-E-Mail erhalten

Checkliste:

1. SMTP_LOGIN_* Variablen korrekt gesetzt?
2. Server neu gestartet nach `.env` -Änderungen?
3. E-Mail-Adresse im Benutzer-Account korrekt?
4. Spam-Ordner prüfen
5. Console-Logs prüfen:

```
bash
```

```
tail -f /var/log/your-app.log | grep "login notification"
```

Keine Update-E-Mail erhalten

Checkliste:

1. `emailReportFrequency` im Profil auf `"daily"` oder `"weekly"` gesetzt?
2. `emailNotifications` ist `true` ?
3. SMTP_UPDATE_* Variablen korrekt gesetzt?
4. Cron-Job läuft? (Vercel Crons nur in Pro/Team)
5. Manuell testen:

```
bash
```

```
curl "http://localhost:3000/api/send-update-emails?frequency=daily&test=true"
```

Monitoring

Console-Logs

```
# Login-Benachrichtigungen
grep "login notification" /var/log/your-app.log

# Update-E-Mails
grep "update email" /var/log/your-app.log

# SMTP-Fehler
grep "SMTP" /var/log/your-app.log
```

Vercel Logs

```
vercel logs --follow
```

E-Mail-Statistiken

API-Aufruf:

```
curl "http://localhost:3000/api/send-update-emails?frequency=daily"
```

Response:

```
{
  "success": true,
  "frequency": "daily",
  "totalUsers": 10,
  "successful": 9,
  "failed": 1,
  "timestamp": "2026-01-08T10:00:00.000Z"
}
```



Zusammenfassung

E-Mail-Typen

Typ	Von	Wann	Benutzer-Kontrolle
Ticket/Subtask	noreply@planbar.dev	Bei Zuweisung/ Statusänderung	<input checked="" type="checkbox"/> Profil: emailNotifications
Login	noreply@planbar.dev	Bei jedem Login	<input checked="" type="checkbox"/> Immer aktiviert
Updates	update@planbar.dev	Täglich/Wöchentlich	<input checked="" type="checkbox"/> Profil: emailReportFrequency

SMTP-Konten

Zweck	E-Mail	Umgebungsvariablen- Präfix
Tickets/Subtasks	noreply@planbar.dev	SMTP_
Login	noreply@planbar.dev	SMTP_LOGIN_
Updates	update@planbar.dev	SMTP_UPDATE_

Wichtige Dateien

Datei	Zweck
.env	SMTP-Konfiguration
lib/email.ts	E-Mail-Templates und Versand-Funktionen
lib/auth.ts	Login-Benachrichtigung (JWT Callback)
app/api/send-update-emails/route.ts	Update-E-Mails API
app/api/test-email/route.ts	Test-E-Mail API

🔗 Links

- **E-Mail-Setup-Anleitung:** EMAIL_SETUP.md
- **Hostpoint SMTP-Dokumentation:** <https://www.hostpoint.ch/> (<https://www.hostpoint.ch/>)
- **Nodemailer Docs:** <https://nodemailer.com/> (<https://nodemailer.com/>)
- **Vercel Cron-Jobs:** <https://vercel.com/docs/cron-jobs> (<https://vercel.com/docs/cron-jobs>)

Status:  **Vollständig implementiert und getestet**

Autor: DeepAgent

Datum: 8. Januar 2026