

Workigom Veritabanı Seeding Rehberi

Bu rehber, Workigom platformunun veritabanına test kullanıcıları ve örnek verileri nasıl ekleyeceğinizi açıklar.

İçindekiler

1. Gereksinimler
2. Railway’de Seed Çalıştırma
3. Yerel Ortamda Seed Çalıştırma
4. Seed İşlemi Sonrası Kontrol
5. Sorun Giderme

Gereksinimler

- Node.js (v18 veya üzeri)
- npm veya yarn
- PostgreSQL veritabanı (Railway üzerinde veya yerel)
- Prisma CLI

Railway’de Seed Çalıştırma

Railway üzerinde deploy edilmiş backend servisimize test verileri eklemek için:

Yöntem 1: Railway CLI ile (Önerilen)

1. **Railway CLI’yi yükleyin** (eğer yoksa):

```
bash
npm install -g @railway/cli
```

2. **Railway’e giriş yapın:**

```
bash
railway login
```

3. **Projenize bağlanın:**

```
bash
railway link
```

Listeden `workigom` projesini ve `backend` servisini seçin.

1. **Seed komutunu çalıştırın:**

```
bash
railway run npm run prisma:seed
```

Veya doğrudan:

```
bash
railway run npx prisma db seed
```

Yöntem 2: Railway Dashboard Üzerinden

1. **Railway Dashboard'a gidin:** <https://railway.app/>
2. **Projenizi açın:** `workigom` projesini bulun
3. **Backend servisini seçin:** `workigom` servisini tıklayın
4. **Settings sekmesine gidin**
5. **Environment Variables bölümünde** `DATABASE_URL` değişkeninin doğru ayarlandığından emin olun
6. **Deploy sekmesine gidin**
7. **Command alanına** şunu yazın:

```
bash
npm run prisma:generate && npm run prisma:migrate:deploy && npm run prisma:seed && npm start
```
8. **Redeploy** butonuna tıklayın

Yöntem 3: Railway CLI ile Doğrudan Komut

```
railway run --service workigom "npx prisma db seed"
```



Yerel Ortamda Seed Çalıştırma

Yerel geliştirme ortamınızda test verileri oluşturmak için:

Adım 1: Veritabanı Bağlantısını Ayarlayın

1. **Backend dizinine gidin:**

```
bash
cd /home/ubuntu/workigom/backend
```
2. **.env dosyasını oluşturun** (eğer yoksa):

```
bash
cp .env.example .env
```
3. **.env dosyasını düzenleyin** ve `DATABASE_URL` değişkenini ayarlayın:

```
env
DATABASE_URL="postgresql://kullanici:sifre@localhost:5432/workigom_db?schema=public"
```

Railway PostgreSQL kullanıyorsanız:

- Railway Dashboard'dan `PostgreSQL` servisini açın
- `Variables` sekmesinden `DATABASE_URL` değerini kopyalayın
- `.env` dosyasına yapıştırın

Adım 2: Bağımlılıkları Yükleyin

```
npm install
```

Adım 3: Prisma Client Oluşturun

```
npm run prisma:generate
```

Adım 4: Migration'ları Çalıştırın

```
npm run prisma:migrate:deploy
```

Veya geliştirme ortamında:

```
npm run prisma:migrate dev
```

Adım 5: Seed Komutunu Çalıştırın

```
npm run prisma:seed
```

Veya doğrudan:

```
npx prisma db seed
```


Seed İşlemi Sonrası Kontrol

Terminal Çıktısını Kontrol Edin

Başarılı bir seed işlemi şöyle görünmelidir:

 Seeding **database**...


 **Admin user** created: admin@workigom.com
 Corporate **user** 1 created: company1@workigom.com
 Corporate **user** 2 created: company2@workigom.com
 Corporate **user** 3 created: company3@workigom.com
 Corporate **user** 4 created: company4@workigom.com
 Individual **user** 1 created: mehmet@example.com
 Individual **user** 2 created: ayse@example.com
 Individual **user** 3 created: ali@example.com
 Individual **user** 4 created: fatma@example.com
 Individual **user** 5 created: can@example.com

 Creating Jobs...


 Job 1 created: Yazılım Geliştirici
 Job 2 created: Garson
 Job 3 created: Depo Elemanı
 Job 4 created: Müşteri Temsilcisi
 Job 5 created: Grafik Tasarımcı

 Creating Applications...


[...]

 Creating Donations...


[...]

 Creating Messages...

[...]

 Creating Notifications...

[...]

 **Database** seeding completed successfully!

 **Summary:**

- 1 **Admin user**
- 4 Corporate users (Employers/Donors)
- 5 Individual users (Job Seekers/Donation Receivers)
- 5 Jobs
- 5 Applications
- 5 Donations
- 4 Messages
- 6 Notifications

 **Default Passwords:**

- **Admin**: admin123
- Corporate **users**: company123
- Individual **users**: user123

Prisma Studio ile Kontrol Edin

```
npm run prisma:studio
```

Tarayıcınızda `http://localhost:5555` adresinde Prisma Studio açılacaktır. Burada oluşturulan kullanıcıları ve verileri görsel olarak inceleyebilirsiniz.

API ile Kontrol Edin

Backend sunucunuz çalışıyorsa, şu endpoint'leri test edebilirsiniz:

```
# Kullanıcıları listele (admin olarak giriş yapmanız gerekebilir)
curl http://localhost:3001/api/users

# İş ilanlarını listele
curl http://localhost:3001/api/jobs

# Bağışları listele
curl http://localhost:3001/api/donations
```

Giriş Yaparak Test Edin

Frontend uygulamasında şu kullanıcılarla giriş yapabilirsiniz:

- **Admin:** admin@workigom.com / admin123
- **Kurumsal:** company1@workigom.com / company123
- **Bireysel:** mehmet@example.com / user123



Veritabanını Sıfırlama ve Yeniden Seed Etme

Veritabanını tamamen temizleyip baştan seed etmek isterseniz:

Dikkat: Bu işlem TÜM verileri silecektir!

```
# Veritabanını sıfırla ve yeniden migrate et
npm run prisma:migrate reset

# Bu komut otomatik olarak seed'i de çalıştırır
```

Veya manuel olarak:

```
# Migration'ları sıfırla
npx prisma migrate reset --skip-seed

# Seed'i tekrar çalıştır
npm run prisma:seed
```



Sorun Giderme

Hata: “Error: P1001: Can’t reach database server”

Çözüm:

- DATABASE_URL değişkeninin doğru ayarlandığından emin olun
- PostgreSQL sunucusunun çalıştığından emin olun
- Ağ bağlantınızı kontrol edin
- Railway’de: Servisin aktif olduğundan emin olun

```
# DATABASE_URL'yi test edin
echo $DATABASE_URL

# Railway'de:
railway variables
```

Hata: “Error: P2002: Unique constraint failed”

Çözüm:

Bu hata, seed verileri zaten veritabanında varsa oluşur. Veritabanını sıfırlayın:

```
npm run prisma:migrate reset
```

Hata: “Command failed: ts-node prisma/seed.ts”

Çözüm:

TypeScript ve ts-node’un yüklü olduğundan emin olun:

```
npm install --save-dev typescript ts-node @types/node
```

Hata: “Error: Cannot find module ‘@prisma/client’”

Çözüm:

Prisma client’ı generate edin:

```
npm run prisma:generate
```

Hata: “Error: Environment variable not found: DATABASE_URL”

Çözüm:

.env dosyasının mevcut olduğundan ve DATABASE_URL değişkeninin ayarlandığından emin olun:

```
# .env dosyasını kontrol edin
cat .env | grep DATABASE_URL

# Yoksa .env.example'dan kopyalayın
cp .env.example .env
```

Railway’de Seed Çalışmıyor

Çözüm:

1. Railway Dashboard’da Settings > Deploy bölümünde **Build Command** ve **Start Command** değerlerini kontrol edin:

Build Command:

```
bash
```

```
npm install && npm run prisma:generate && npm run build
```

Start Command:

```
bash
```

```
npm run prisma:migrate:deploy && npm start
```

1. Seed'i manuel olarak çalıştırmak için Railway CLI kullanın:

```
bash
```

```
railway run npm run prisma:seed
```

2. Logs'u kontrol edin:

```
bash
```

```
railway logs
```



Ek Kaynaklar

- [Prisma Seeding Dökümanı](https://www.prisma.io/docs/guides/database/seed-database) (https://www.prisma.io/docs/guides/database/seed-database)
- [Railway Dökümanı](https://docs.railway.app/) (https://docs.railway.app/)
- [Workigom Test Kullanıcıları](#) (./TEST_USERS_CREDENTIALS.md)



İpuçları

1. **Seed işlemini düzenli olarak çalıştırın:** Geliştirme sırasında veritabanını sık sık sıfırlayıp seed edebilirsiniz.
2. **Kendi test verilerinizi ekleyin:** `prisma/seed.ts` dosyasını düzenleyerek kendi test verilerinizi ekleyebilirsiniz.
3. **Production'da dikkatli olun:** Production ortamında seed çalıştırmadan önce mevcut verileri yedekleyin!
4. **Prisma Studio kullanın:** Veritabanı verilerini görsel olarak incelemek ve düzenlemek için Prisma Studio çok kullanışlıdır.



Yardım

Sorun yaşıyorsanız:

1. Logs'u kontrol edin
2. Database bağlantısını test edin
3. Environment variables'ı kontrol edin
4. GitHub Issues'da sorun açın