



Railway Next Steps - Action Guide

Status:  Fix has been pushed to GitHub

Commit: `a32e38a` - “Include Prisma migrations in version control”

What Changed: Added `backend/prisma/migrations/` to git repository

🎯 Immediate Actions

Step 1: Verify Files in GitHub (2 minutes)

Go to your repository and confirm migrations are present:

URL: <https://github.com/volkanakbulut73/workigom/tree/master/backend/prisma/migrations>

You should see:

-  `migration_lock.toml`
-  `20251022231535_exit/` folder
-  `migration.sql` inside

If you don't see these files, the push might not have completed. Run:

```
cd /home/ubuntu/workigom
git push origin master
```

Step 2: Check Railway Auto-Build (1 minute)

Railway should automatically detect the new commit and start building.

1. **Go to Railway Dashboard:**

[https://railway.app/project/\[your-project-id\]](https://railway.app/project/[your-project-id])

2. **Check Deployments Tab:**

- Look for a new deployment with commit `a32e38a`
- Status should show “Building” or “Deploying”

3. **If NO new deployment appears:**

- Continue to Step 3 to manually trigger

Step 3: Manually Trigger Rebuild (if needed)

If Railway didn't auto-build, trigger manually:

Option A: Via Dashboard (Recommended)

1. Go to Railway Dashboard
2. Select `workigom` service (backend)
3. Click “**Deployments**” tab

4. Click “⋮” (three dots) on latest deployment

5. Click “**Redeploy**”

Option B: Via Empty Commit

```
cd /home/ubuntu/workigom
git commit --allow-empty -m "trigger: rebuild with prisma migrations"
git push origin master
```

Step 4: Monitor Build Logs (5-10 minutes)

Watch the Railway build logs for these indicators:

Success Indicators:

1. " WORKIGOM BACKEND DOCKERFILE - If you see this..."
 Correct Dockerfile is being used
2. " COPY backend/prisma ./prisma/"
 Prisma directory successfully copied
3. " RUN npm run prisma:generate"
 Prisma Client generated successfully
4. " Starting Workigom Backend..."
 Application starting
5. " Running Prisma migrations..."
 Migrations **running**
6. " Starting server..."
 Server started successfully

If You See Errors:

Error: “prisma: not found”

- Check that `DATABASE_URL` environment variable is set

Error: “Can't reach database”

- Verify PostgreSQL service is running
- Check `DATABASE_URL` format: `postgresql://user:pass@host:port/dbname`

Error: Still says “/backend/prisma: not found”

- Verify commit `a32e38a` is being deployed
- Check GitHub to confirm files are present
- Try clearing Railway build cache (Settings → Advanced → Clear Cache)

Step 5: Verify Deployment (2 minutes)

Once build completes successfully:

Test Health Endpoint:

```
# Replace with your actual Railway URL
curl https://workigom-production.up.railway.app/api/health
```

Expected response:

```
{
  "status": "ok",
  "timestamp": "2025-10-23T21:00:00.000Z"
}
```

Test Database Connectivity:

```
curl https://workigom-production.up.railway.app/api/jobs
```

Should return job listings (empty array `[]` is also valid).

Railway Settings to Verify

Double-check these settings in Railway Dashboard:

Service: `workigom` (Backend)

1. Source Repository Settings

```
Repository: volkanakbulut73/workigom
Branch: master  Should show "connected to production"
Root Directory: backend/  Can be "backend/" or "backend"
```

How to check:

1. Railway Dashboard → `workigom` service
2. Click “**Settings**” tab
3. Scroll to “**Source Repo**” section

2. Build Settings

```
Builder: DOCKERFILE
Dockerfile Path: backend/Dockerfile
```

How to check:

1. Railway Dashboard → `workigom` service
2. Click “**Settings**” tab
3. Scroll to “**Build**” section

Note: Your `railway.toml` already configures this:

```
[build]
builder = "DOCKERFILE"
dockerfilePath = "backend/Dockerfile"
```

3. Environment Variables

Required variables:

- DATABASE_URL - PostgreSQL connection string
- JWT_SECRET - Your JWT secret
- PORT - Should be 3001
- NODE_ENV - Should be production

How to check:

1. Railway Dashboard → `workigom` service
2. Click “**Variables**” tab
3. Verify all required variables are set



Current Git Status

Latest Commit: a32e38a
Branch: master
Remote: github.com:volkanakbulut73/workigom.git

Files Added in Last Commit:

- backend/prisma/migrations/migration_lock.toml
- backend/prisma/migrations/20251022231535_exit/migration.sql

Files Modified:

- backend/.gitignore (removed prisma/migrations/ exclusion)

What Should Happen Now

Expected Flow:

1. GitHub → Railway detects **new** commit (a32e38a)
↓
2. Railway → Clones repository with migrations
↓
3. Docker Build → Starts using backend/Dockerfile
↓
4. COPY backend/prisma → Successfully copies migrations
↓
5. Prisma Generate → Creates Prisma Client
↓
6. Build Complete → Creates production image
↓
7. Container Start → Runs startup script
↓
8. Prisma Migrate → Executes "npx prisma migrate deploy"
↓
9. Server Start → Express server starts **on** port 3001
↓
10. Health Check → Passes
↓
11. ✓ DEPLOYMENT SUCCESS

Troubleshooting

Issue: Railway still using old Dockerfile

Symptoms:

- Build logs don't show " WORKIGOM BACKEND DOCKERFILE"
- Still getting errors about missing files

Solution:

1. Check `railway.toml` in root:

```
toml
[build]
builder = "DOCKERFILE"
dockerfilePath = "backend/Dockerfile"
```

2. Verify Root Directory is set to `backend/` OR empty

3. Clear Railway build cache:

- Settings → Advanced → Clear Build Cache

Issue: Migrations not running

Symptoms:

- Build succeeds but database tables don't exist
- Application errors about missing tables

Solution:

1. Check Railway logs for "Running Prisma migrations..."
2. Verify `DATABASE_URL` is set correctly
3. Manually run migrations in Railway shell:

```
bash
  npx prisma migrate deploy
```

Issue: Build succeeds but app crashes

Symptoms:

- Build completes successfully
- Container starts but immediately crashes
- Health check fails

Solution:

1. Check environment variables (especially `DATABASE_URL`)
 2. View runtime logs in Railway (not build logs)
 3. Verify database is accessible from Railway's network
-



Getting Help

If issues persist after following this guide:

1. **Check Full Diagnostic Report:** See `PRISMA_DIAGNOSTIC_REPORT.md`
 2. **Railway Logs:** Copy full build and runtime logs
 3. **GitHub Repository:** Verify files at <https://github.com/volkanakbulut73/workigom>
 4. **Railway Support:** <https://help.railway.app>
-



Success Checklist

- [] Files visible in GitHub repository
 - [] Railway detected new commit
 - [] Build started automatically (or manually triggered)
 - [] Build logs show correct Dockerfile marker
 - [] Prisma files copied successfully
 - [] Prisma client generated
 - [] Build completed without errors
 - [] Container started
 - [] Migrations deployed
 - [] Server started on port 3001
 - [] Health check endpoint returns 200 OK
 - [] Application is accessible via Railway URL
-

Expected Timeline

| Step | Duration | Status |
|------------------------|---------------------|--------|
| Push to GitHub | Completed | ✓ |
| Railway detects commit | 30 seconds | ⌚ |
| Build starts | Immediate | ⌚ |
| Docker build | 3-5 minutes | ⌚ |
| Container start | 30 seconds | ⌚ |
| Migration deploy | 10-30 seconds | ⌚ |
| Health check | 5 seconds | ⌚ |
| Total | ~5-7 minutes | ⌚ |

Quick Reference

Verify Files in GitHub

```
https://github.com/volkanakbulut73/workigom/tree/master/backend/prisma/migrations
```

Railway Dashboard

```
https://railway.app
→ Select your project
→ Select "workigom" service
→ Check "Deployments" tab
```

Test Health Endpoint (after deployment)

```
curl https://[your-railway-url].up.railway.app/api/health
```

Force Rebuild Command

```
cd /home/ubuntu/workigom
git commit --allow-empty -m "trigger: rebuild"
git push origin master
```

Last Updated: October 23, 2025

Status:  Ready for Railway deployment

Next Step: Monitor Railway dashboard for automatic rebuild
